SA 1/87

ÖS 55 sfr 6,50 DM 6,50

Compute mit

COMMODORE

C=16

Heimcomputer

C=116 / plus 4

Sonderheft 1/87

Diesmal:

Wettbewerb
plus 4 & Floppy
und viele andere Preise
zu gewinnen!

Sound & Grafik:
Alles über den
TED-CHIP!

TOPGAMES
in Maschinensprache

Jede Menge
Utilities!

Wichtige Systemroutinen
und
vieles mehr.

Turbo Tape

DAS GOLDENE KINGSOFT ANGEBOT

für Ihren Commodore C-16:

DAS GROSSE C-16 BUCH

Hier erfahren Sie alles, was Sie brauchen, um die Möglichkeiten Ihres C-16 voll auszunutzen. Sämtliche wichtigen Bereiche, wie z.B. Grafik, Sound, Maschinensprache werden ausführlich behandelt und mit vielen Beispielen erläutert. Dieses Buch ist für jeden Commodore-16 Besitzer unentbehrlich.

Einzelpreis: 29.-

SE DESIGN

JOYSTICK

Original Commodore-Joystick im eleganten und handlichen Design für viele Stunden Spielspaß; steigert die Freude bei praktisch allen Spielen enorm. Direkt anschlußfertig für Ihren C-16.

Einzelpreis: 19.-

PLUS-PAKET

Die ideale Spielesammlung für alle Einsteiger, bestehend aus 4 absoluten Top-Programmen (siehe auch Rückseite):

GRANDMASTER, spielstarkes Schachprogramm mit viel Komfort
TOM, faszinierendes Arcade-Adventure mit 178 (!) Bildern
GALAXY, schnelles und abwechslungsreiches Weltraumspiel
GHOST TOWN, spannendes Grafik-Adventure für die ganze Familie

Einzelpreis: 29.-

Alle 3 Teile zusammen statt ~~77.-~~ nur 49.-

Sie sparen 28.-

SPITZEN-SOFTWARE
KINGSOFT
MADE IN GERMANY

F. SCHÄFER
SCHNACKEBUSCH 4
5106 ROETGEN
☎ 02408/5119

EDITORIAL

Lieber Leser!

Eine kleine Verwirrung möchte ich gleich zu Beginn geraderücken. In der Ankündigung dieses Sonderheftes in "Compute mit" 12/86 haben wir dieses Heft als Ausgabe 4 bezeichnet. Um aber den Zeitschriftenhandel nicht zu verwirren, trägt diese Ausgabe korrekterweise die Nummer 1/87. Natürlich handelt es bei diesem Heft um die Ausgabe, die noch im Dezember erscheint, und daher möchte ich auf diesem Wege allen unseren Lesern nicht nur ein frohes Fest, sondern ganz besonders ein erfolgreiches Jahr 1987 wünschen (auf daß die Hardware- und Softwarepreise weiter fallen).

Das vorliegende Sonderheft ist in mehreren Punkten ein ganz besonderer Leckerbissen. Eine dieser Spezialitäten dieses Heftes, ist der Schwerpunkt über die Grafik- und Soundprogrammierung mit dem TED-Chip des C 16/116/plus 4, die unseres Wissens nach an keiner anderen Stelle so ausführlich beschrieben wird, wie in diesem Heft. Den C16/116/plus 4-Besitzern wird also eine hervorragende Programmierunterstützung in die Hand gegeben, wie schon mit dem Sonderheft 3, in dem der Genesis-Super-Assembler veröffentlicht wurde, der momentan auf dem gesamten C-16-Softwaremarkt absolut konkurrenzlos dasteht.

Als Folge ständig verbesserter Programmiermöglichkeiten können wir in diesem Heft Programme exzellenter Qualität anbieten. Unsere Stammautoren schreiben mittlerweile ihre Programme vornehmlich in Maschinensprache, denn nur so läßt sich der Speicherplatz des C-16/116 voll ausnutzen. Am Ende haben die technischen Einschränkungen dieses Computersystems einen entscheidenden Vorteil: man wird dazu gezwungen, in die Funktionsweise des Computers Einblick zu nehmen und kann sich nicht, wie bei den neuen großen Maschinen (AMIGA und ATARI ST) auf komfortable Programme verlassen. Immer mehr Besitzer des C 16 jedenfalls sind zu hervorragende Programmierern der Maschinensprache herangewachsen. Wann gehören auch Sie dazu?

Und: wenn Sie nicht den Ehrgeiz dazu haben, können Sie sicher sein, daß Sie für sehr wenig Geld allerhand erhalten haben. Auch wenn das, was Sie mit Ihrem Computer machen können, nicht in allen Dingen so toll aussieht, wie mit einem 16-Bitter, der aber auch das Zehnfache kostet.

Uwe Knierim
Chefredakteur

report

Neue Software-Produkte 4
Paperboy in Vorbereitung 5
News 39

tips & tricks

Jetzt ziehen wir alle Register 23
Datamaker 25
Auto-Old-Routine 27
Double Screen 28
2 x Laufschrift 31
Screen Editor 42
dload - erweitert 44
Speichersplit 64 K 45
Turbo-Tape 46

programme

Kundenrechnungen 38
Riesenlabyrinth 49
Crazy Worms 51
Convoy 55
Rich Freddy 60

... und

Checksummer 6
Klartexttabelle 7
Kurs: Der Tedchip, Grafik und Sound 8
Wettbewerb 33
Softwareservice 34
Kleinanzeigen 64

IMPRESSUM

"Compute mit" Sonderheft
erscheint im
Tronic-Verlag, Am Stad 35, 3440 Eschwege
Tel.:05651/3 00 11

Redaktion:

Axel Credé (verantwortlich)
Chefredakteur: Uwe Knierim
Redakteure: Manfred Kleimann, Bernd Zimmermann, Frank Brall, Ottfried Schmidt, Thomas Brandt

Gesamtherstellung:

Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Anfragen nicht an den Vertrieb oder Druckerei, sondern nur an den Verlag!

Vertrieb

Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Str. 20
Telefon: 06121/2660

Erscheinungsweise:

Erstverkaufstag von "Compute mit" jeweils Mitte des Monats.

Urheberrecht:

Alle in "Compute mit" veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.

Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen und Bezeichnungen frei von Schutzrechten sind.

Bezugspreis:

Sonderheft 6,50 DM
Normalausgabe Einzelheft 3,80 DM
Abonnement: Inland 42,-- DM im Jahr (12 Ausgaben)
Ausland (Europa): 52,-- DM
ohne Kassetten!

Programmierabteilung:

Montag – Freitag von 14-16 Uhr
Tel. (05651) 30013

Autoren, Manuskripte:

Der Verlag nimmt Manuskripte zur Veröffentlichung gerne entgegen.

Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,-- DM pro abgedruckte Seite einverstanden sind.

Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Alle Einsendungen müssen frei von Rechten Dritter sein. Dies muß mit der Unterschrift des Einsenders bestätigt werden.

Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:

Kopierfähige Kassette oder Diskette mit dem Programm (Computerbezeichnung nicht vergessen!), von Drucker ersteltes Listing (keine Schreibmaschinenlistings!), evtl. Bildschirmfotos oder Hardcopies mit Demonstrationsbeispielen und ausführliche Programmbeschreibung (Erklärung der programmtechnischen Besonderheiten, Spielverlaufbeschreibung). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

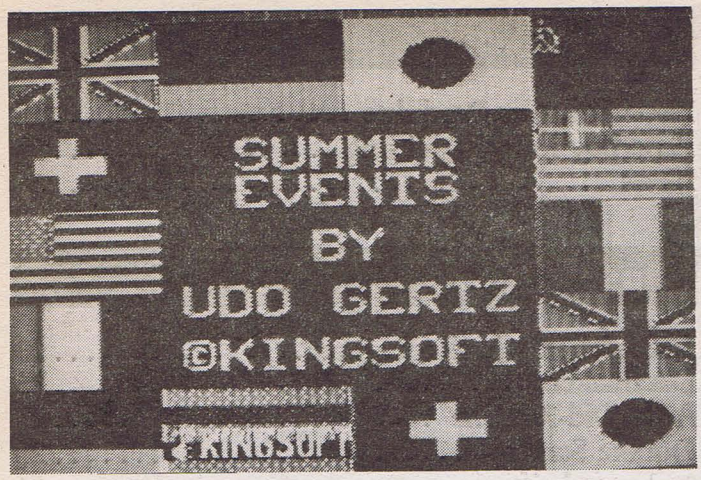
Anzeigenpreise:

Bitte Mediaunterlagen anfordern.

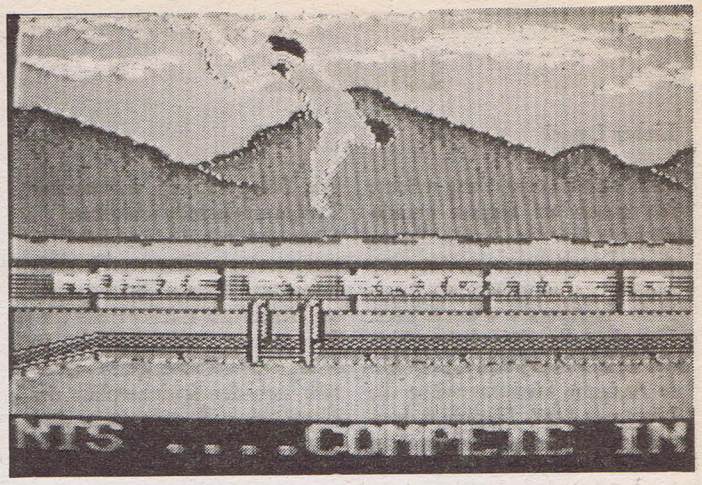
Anzeigenverwaltung:

Anzeigenleiter: Hartmut Wendt

Tronic-Verlag GmbH
Am Stad 35, 3440 Eschwege
Telefon: 05651/30011
Telefax: 05651/30011



Eröffnungsbild »Sommer-Olympiade«



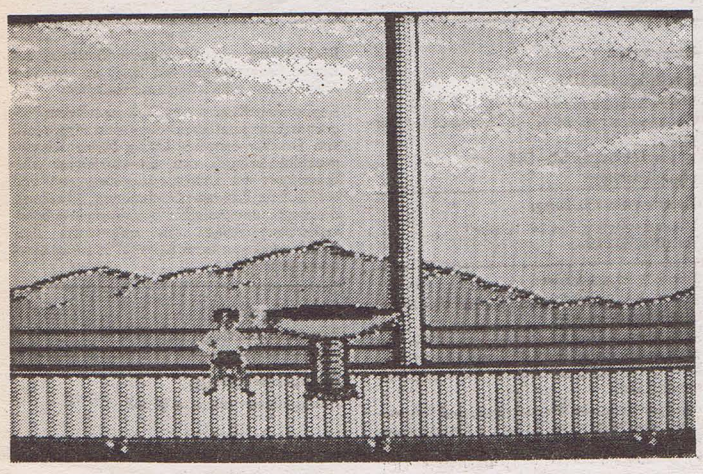
Ein Turmspringer in seinem »nassen« Element!

Neue Software-Produkte von C-16/116 und Plus 4 Spezialisten aus Deutschland

Die Fa. Kingsoft entwickelt sich im deutschen Software-Markt zum Programm-Spezialisten für die »kleinen Commodore«!

Nachdem das Jahr 1986 für die Fa. Kingsoft von der produktionstechnischen Seite einen guten und erfolgreichen Abschluß gefunden hat, will man auch 1987 wieder mit neuen Produkten glänzen. Der absolute Software-Hit soll nach dem Willen der Macher das Programm "Sommer Olympiade" werden. Wir haben uns natürlich mit der Fa. Kingsoft in Verbindung gesetzt und nach dem Stand der Dinge gefragt. So war zu erfahren, daß die ersten 3 Disziplinen der "Sommer Olympiade" bereits fertiggestellt sind und an den drei weiteren fieberhaft gearbeitet wird. Es war uns vergönnt, verschiedene Programmausschnitte und -folgen zu testen. "Sommer Olympiade" ist vom Ablauf her ähnlich dem der "kalten Spiele" aufgebaut. So können Sie auch hier wieder zwischen sechs verschiedenen Disziplinen wählen (u.a. Radfahren, Stabhochsprung, Turmspringen, Wildwasserkanu usw.). Eine Eröffnungszeremonie, mit Wahl der Landesfarben und der Hymnen ist dabei ebenso vorgesehen, wie die Wahl der Mitspieler (insgesamt 4). Bei der Disziplin "Radfahren" ist eine lebensechte Animation verwirklicht worden und wir waren schon etwas überrascht, daß es dem Programmierer wieder gelungen ist, eine so bestechend gute Grafik zu erstellen. Auch bei der Disziplin "Turmspringen" könnte man meinen, man befände sich in der Umgebung eines echten Schwimmbades und ist Teilnehmer eines spanndenen Wettkampfes, bei dem es darum geht, die erforderlichen Höchstnoten für jeden Sprung zu erzielen. Natürlich können wir hier noch keinen vollständigen Testbericht veröffentlichen, aber bei den wenigen Ausschnitten und Spielabläufen die wir gesehen haben, kann man schon jetzt vermuten, daß auch das Programm "Sommer Olympiade" ein echter Hit werden wird. Es dürfte seinem so erfolgreichen Nachfolger "Winter Olympiade" mit Sicherheit in nichts nachstehen. Die Fortsetzung dieses Sportspiels ist ein Muß für jeden Anwender und sollte deshalb schon jetzt auf Ihrem Software-Wunschzettel für 1987 stehen. Mit der Veröffentlichung des Programmes ist übrigens im Januar '87 zu rechnen.

***



Die Eröffnungsfeier mit entzünden der olympischen Flamme im Stadion!

Weitere Software-Neuheiten

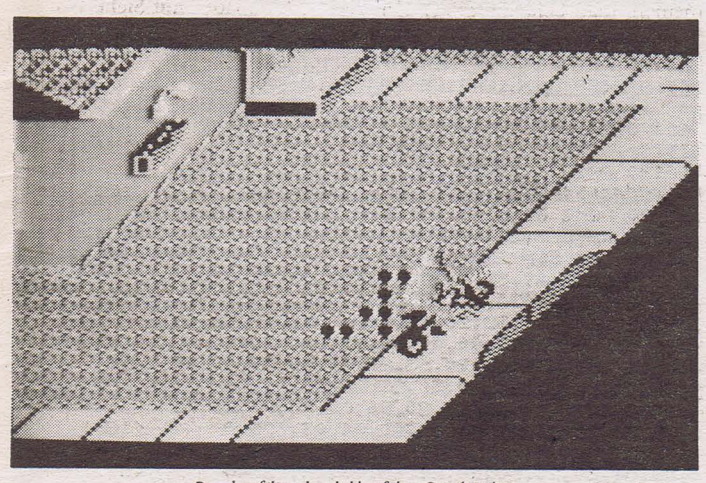
Zwei weitere Software-Neuheiten der Fa. Kingsoft sind die Programme:

Brigdehead und Karate King

Bei **Brigdehead** handelt es sich um eine gelungene Adaption der Green-Beret-Spielidee. Sie müssen hier nach alter "Rambo"-Manier

über das Spielfeld hasten und alles niedermachen was sich Ihnen in den Weg stellt. Für diese Aufgabe stehen Ihnen natürlich verschiedene Waffen als Hilfsmittel zur Verfügung. Sie können die Angreifer per Hand, mit Pistole oder Gewehr bis hin zur Handgranate zur Strecke bringen. Sollten Sie einmal von einem Ihrer Gegener ins Jensseits befördert worden sein, beginnt das Spiel nicht von vorn, sondern wird an der Stelle fortgesetzt, an der Sie zuletzt gekämpft haben.
Insgesamt ist dieses Programm in acht verschiedene Level mit jeweils 9 unterschiedlichen Bildern unterteilt. Für diesen programmiertechnischen Gewaltakt sind aber auch volle 30K-Bytes Speicherplatz nötigt, so daß Sie schon einen erweiteren Rechner oder den Plus 4 verwenden müssen. Nach Auskunft des Herstellers soll es auch eine Version für die Grundgeräte geben, jedoch wird diese dann mit Sicherheit nicht alle hier vorgestellten Schwierigkeiten und Rafinessen enthalten. Einzelkämpferfanatiker und andere Kämpfernaturen werden an diesem Action-Spiel ihre helle Freude haben.
Das zweite vorgestellte Programm **"Karate King"** läßt schon vom Titel her keine Zweifel aufkommen, um welche Art von Spiel es sich hierbei handelt. Hier müssen alle Freunde der japanischen Kampfsportart Kung-Fu ihre Fähigkeiten unter Beweis stellen. Dabei sind schnelle Reaktionen, ein gutes Auge für die Situation und die entsprechende Kampfstärke gefragt. Sie müssen ihren Gegner mit Hilfe von Fußtritten, Faustschlägen und anderen den Regeln entsprechenden Kampfhandlungen auf die Matte zwingen. Im Hintergrund wacht der "Meister" über das Kampfgeschehen und gibt am Ende des Schlagabtausches den Sieger bekannt. Die gute Grafik und die spannende Spielanlage lassen vermuten, daß dieses Spiel schnell seine Liebhaber finden wird. Wenn Sie sich also angesprochen fühlen, setzten Sie sich mit dem Hersteller in Verbindung. Wir können dieses Programm allen Kung-Fu Freunden empfehlen und wünschen Ihnen viel Spaß!

»Paperboy«-Adaption befindet sich in Vorbereitung



Paperboy fährt schon bald auf dem C-16/116/+4

Die interessanteste Nachricht erreichte uns dann noch kurz vor Redaktionsschluß. Die Fa. Kingsoft hat von dem englischen Software-Haus Elite den Auftrag zur Adaption des C-64 Erfolges "Paperboy" erhalten. So ist in den nächsten Wochen mit der Veröffentlichung des bekannten Spielhallenrenners auch für die C-16/116 und Plus 4 Besitzer zu rechnen. Uns kann es nur recht sein, daß auch diese Systeme immer mehr an Bedeutung gewinnen und die Softwarehäuser erkannt haben, daß hier ein außergewöhlich guter Rechner seine Liebhaber gefunden hat. Man darf also auf die Zukunft gespannt sein.

Anwendersoftware für den Computerpraktiker

Bei Kingsoft hat man sich nicht nur auf die Programmierung von Spielen beschränkt, man hat auch im Anwender-Bereich eine Vielzahl von Programmen entwickelt. Ein Blick auf unser Bild zeigt uns insgesamt 6 verschiedene Anwenderprogramme
Grafik Designer
Music Master
Turbo Tape
Paintbox
Micro Kalk und
Mikro Datei •
die den User bei seiner täglichen Arbeit mit dem Computer unterstützen sollen. Da wir Turbo Tape bereits vorgestellt haben, möchten wir hier nicht versäumen, ein paar Worte über die anderen Programme zu verlieren. Als da wäre zunächst das Programm **"Grafik-Designer.** Mit Hilfe eines Editors können Sie hier Ihren eigenen Zeichensatz erstellen. Dabei müssen Sie auch nicht auf so aufwendige Sachen wie Scrollen, Spiegeln, Invertieren, Drehen, um nur einige zu nennen, verzichten. Selbst eine Animation von mehreren Zeichen ist möglich. Sie sehen also ein äußerst nützliches Programm. Aber auch **Paintbox"** ein Mal- und Zeichenprogramm in 121 Farben ist eine durchaus interessante Anwendung. Acht verschiedene Pinsel bieten unterschiedliche Anwendungsmöglichkeiten. So kann man z.B. Flächen füllen, Linien ziehen, Kreise, Rechtecke und Scheiben zeichen u.v.m. Die Steuerung erfolgt über Joystick oder Tastatur und im Test erwies sich die Handhabung als äußerst einfach und hinterließ einen positiven Eindruck.
Mit Musikmaster bekommt der C-16/116/Plus4 - User ein Programm zur Verfügung gestellt, daß einen ebenso guten Musiker aus seinem System macht, wie man es von dem "Großen Bruder C-64" gewohnt ist. Dieses Musikprogramm verwandelt Ihr System in einen fantastischen Synthesizer und Sequenzer. So können Sie eigene Musikstücke komponieren, abspeichern und später in selbsterstellte Programme einbauen. Eine tolle Sache für jeden Musik-Fan!
"Last but not least" sind da noch die beiden Programme **"Micro Datei"** und **"Mikro Kalk"**. Micro-Datei ist ein Dateiverwaltungsprogramm mit dessen Hilfe Sie alle Arten von Daten verarbeiten können. Dabei ist ein
* freier Maskenaufbau,
* das Sortieren nach beliebigen Kriterien
* das Suchen von beliebigen Ausdrücken usw.

garantiert. Die leichte Bedienbarkeit und der übersichtliche Aufbau sind dabei von besonderer Bedeutung.
Mit "Mikro-Kalk" liegt ein leitungsstarkes Tabellenkalkulationsprogramm vor, welches die täglich anfallenden Arbeiten (Haushaltskasse, Einkauf-/Verkauf-Erlös usw.) mit Pravour bewältigt. Vielfältige mathematische Funktionen, das Einfügen/Löschen von Spalten/Zeilen sowie das Kopieren von Formeln und Werten sind nur einige Vorzüge dieses Programms.
Jeder der bei seiner täglichen Arbeit mit Daten und Zahlen konfrontiert wird, sollte diese Angebote genau studieren und das für Ihn passende bei seinen Problemlösungen zu Hilfe nehmen.

Weiter S. 39

Der Checksummer für alle C-16/116 – oder Plus 4-Besitzer!

60"

Die Arbeitsweise des C-16/116 Checksummers entspricht weitgehend der VC-20- und C-64-Version und braucht deshalb nicht näher erklärt zu werden. Da auch der C-16/116 nicht den größten Speicher besitzt, wurde hier die Methode der VC-20-Version gewählt, und das Maschinenprogramm in den Kassettenpuffer gelegt. Dies hat den Vorteil, daß kein Basic-Speicher verlorengeht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchgeführt werden dürfen, da diese unweigerlich zum Systemabsturz führen. Um die Kassettenoperationen wieder zuzulassen, muß der Checksummer durch die beiden folgenden SYS-Aufrufe abgeschaltet werden:

SYS 62158 : SYS 33047

Aktiviert wird der Checksummer mit **SYS 818.** Die Funktion und Arbeitsweise entnehmen Sie bitte der C64-Checksummer-Version (siehe Ausgabe Computronic Nr. 3). Eine Ausnahme bei C-16/116 sind die zusätzlichen Steuerzeichen **'FLASHON' (CTRL-),** und **'FLASHOFF' (CTRL.)** welche ebenfalls durch Klartext ersetzt werden (s. Tabelle Tastenbezeichnungen).

Arbeitsweise und Aufbau unseres Checksummers:
Unser Checksummer besteht aus einem kleinen Maschinenprogramm welches als Basic-Loader abgedruckt ist.

Tippen Sie diesen Loader ein und speichern ihn auf Kassette oder Diskette, denn Sie können ihn zukünftig immer wieder benutzen.

Der Start erfolgt durch den Befehl „RUN". Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC ...". Der Checksummer ist nun aktiv und man kann ein beliebiges Tronic-Listing eingeben. Vorher sollte der Checksummer mit dem Befehl „NEW" gelöscht werden. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem Schirm eine Prüfzahl. Vergleichen Sie diese mit der Zahl im Heft hinter der Zeile. Stimmt die Zahl überein, ist die Zeile richtig eingegeben.

Auf die Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.

Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern, Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben.

Für alle, die nicht gerne abtippen, ist der Checksummer unter folgender Bestellnummer zu beziehen:

OV 10 K Kassette 10 DM
OV 10 D Diskette 15 DM

Achtung: nach dem Starten des Checksummers muß noch »NEW (RETURN)« eingegeben werden.

```
1 REM ********************************   <4>
2 REM *                              *   <99>
3 REM *   C16-CHECKSUMMER            *   <140>
4 REM *                              *   <101>
5 REM *   COPYRIGHT BY FRANK BRALL   *   <182>
6 REM *                              *   <103>
7 REM *   (C) 18.04.86               *   <181>
8 REM *                              *   <105>
9 REM ********************************   <12>
10 PRINT"{CLEAR DOWN SPACE2}◆◆◆◆{SPACE2}
CHECKSUMMER{SPACE}QC{SPACE}1.0{SPACE2}◆◆
◆◆{DOWN}"                                <103>
11 PRINT"{SPACE2}◆◆{SPACE2}COPYRIGHT{SPA
CE}FRANK{SPACE}BRALL{SPACE3}◆◆{DOWN}"    <110>
12 PRINT"{SPACE8}FUER{SPACE}TRONIC-VERLA
G"                                       <196>
60000 DIMH(75):FORI=0TO9                 <123>
60010 H(48+I)=I:H(65+I)=I+10:NEXT        <145>
60020 FORI= 818 TO 1010 :READA$          <151>
60030 H=ASC(LEFT$(A$,1))                 <234>
60040 L=ASC(RIGHT$(A$,1))                <253>
60050 D=H(H)*16+H(L):S=S+D:POKEI,D       <13>
60060 A=A+1:IFA<20THENNEXT:A=-1          <97>
60070 READV:Z=Z+1:IFV=STHEN60085         <177>
60080 PRINT"DATAFEHLER{SPACE}IN{SPACE}ZE
ILE{SPACE}:";60200+Z:END                 <39>
60085 IFA<0THEN60100                     <116>
60090 S=0:A=0:NEXT                       <246>
60100 PRINT"{DOWN2}"                     <172>
60110 PRINT"CHECKSUMMER{SPACE}EIN{SPACE}
={SPACE}SYS{SPACE}818{DOWN}"             <140>
60120 PRINT"CHECKSUMMER{SPACE}AUS{SPACE}
={SPACE}SYS{SPACE}62158+SYS{SPACE}33047{
DOWN}"                                   <122>
60130 PRINT"{DOWN}SAVE/LOAD{SPACE}IST{SP
ACE}WAEHREND{SPACE}DER{SPACE}CHECK-"     <61>
60140 PRINT"SUMMER{SPACE}AKTIV{SPACE}IST
,{SPACE}NICHT{SPACE}MOEGLICH{SPACE}!"    <221>
60150 PRINT"{DOWN}BEACHTEN{SPACE}SIE{SPA
CE}DIE{SPACE}HINWEISE{SPACE}IN{SPACE}DEN
"                                        <189>
60170 PRINT"HEFTEN{SPACE}COMPUTE{SPACE}M
IT{SPACE}UND{SPACE}COMPUTRONIC"          <234>
60180 PRINT"{DOWN}VIEL{SPACE}SPASS{SPACE
}!{SPACE}(AUTOR:F.BRALL/6443SONTRA)      <116>
60190 POKE 814,147:POKE815,251:POKE 816,
147:POKE 817,251                         <109>
60195 SYS 818:END                        <177>
60201 DATA A9,3D,8D,02,03,A9,03,8D,03,03
,60,A2,FF,86,3A,20,5A,88,86,3B, 1851     <56>
60202 DATA 84,3C,20,73,04,AA,F0,EF,90,09
,20,53,89,20,79,04,4C,D9,8B,20, 2018     <28>
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D
,E4,03,8D,E5,03,8D,E6,03,18,A5, 2076     <40>
60204 DATA 14,65,15,8D,E6,03,A0,FF,C8,B1
,3B,F0,2C,C9,22,D0,0A,AD,E4,03, 2508     <18>
60205 DATA 49,01,8D,E4,03,B1,3B,AE,E4,03
,D0,04,C9,20,F0,E4,EE,E5,03,AE, 2644     <87>
60206 DATA E5,03,18,B1,3B,6D,E6,03,8D,E6
,03,CA,D0,F4,4C,76,03,38,20,39, 2204     <107>
60207 DATA D8,8E,E2,03,8C,E3,03,A2,00,A0
,00,18,20,39,D8,A9,5B,20,D2,FF, 2365     <115>
60208 DATA A9,00,AE,E6,03,20,5F,A4,A9,5D
,20,D2,FF,20,08,FB,20,08,FB,AC, 2380     <169>
60209 DATA E3,03,AE,E2,03,18,20,39,D8,A2
,00,86,FF,4C,36,87,00,00,00,00, 1778     <133>
60210 DATA 00,00,00,00,00,00,00,00,00,00
,00,00,00, 0                             <104>
```

Neu für alle Commodore Rechner C 64/VC 20 und C 16

Auf vielfachen Leserwunsch haben wir uns entschlossen, unser Druckverfahren für Commodore-Listings zu ändern.
Steuerzeichen sowie **alle Grafikzeichen** werden zukünftig durch Klartext ersetzt. Der Ausdruck ist kompatibel zum bisherigen Druckverfahren, ein neuer Checksummer ist aus diesem Grund **nicht** erforderlich.

Eingabehinweise für alle Commodore Rechner!

Wer sich schon etwas näher mit den Commodore Rechnern befaßt hat, wird sicherlich wissen, daß der Grafiksatz des C64 sehr viele ähnliche Symbole enthält. Aus diesem Grund ist es oft sehr schwer, einzelne Zeichen, wie zum Beispiel horizontale Linien, voneinander zu unterscheiden. Besonders macht sich dieser Nachteil in Listings bemerkbar, welche die Bildschirmgrafik mit PRINT-Zeilen aufbauen.
Bisher haben wir in unserem Ausdruck lediglich die Steuersequenzen wie CTRL-RVS ON oder CTRL-BLUE durch Klartext ersetzt, ab dieser Ausgabe werden jedoch auch alle Grafikzeichen durch ein Schlüsselcode ersetzt. Durch diese Methode sind Verwechselungen praktisch völlig ausgeschlossen.

Wie arbeitet unser Klartext?

Wer schon nach dem alten Verfahren Programme eingegeben hat, wird sicherlich keine Schwierigkeiten mit der Umstellung haben.
Für die neu hinzugekommenen Leser erkläre ich jedoch das Verfahren noch einmal grundlegend.
Alle sogenannten Steuerzeichen (z.B. eine Farbe) sowie ein Grafikzeichen, werden in unseren LISTINGS durch ein CODEWORT, welches die Taste bzw. Tastenkombination kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin (z.B. Computronic o. Compute Mit). Das folgende Beispiel zeigt den Unterschied zwischen unserem alten und dem neuen Druckverfahren:

```
  C64 LISTING MIT CHECKSUMMEN (C V1.0)

10 REM BISHERIGER AUSDRUCK                    <118>
20 PRINT"{DOWN4}┌──────────┐"                <52>
30 PRINT"♠│──┴─│ │ │ │ │ │"                  <85>
40 PRINT"└┬┴┬┴┬┴
```

Die Möglichkeiten des TED

Ein kompletter Kurs in Grafik- und Soundprogrammierung

C 16/116 plus 4

Der im C 16/116 und Plus/4 eingebaute TED-Chip ist ein vielseitiges Multitalent mit erstaunlichen Fähigkeiten. Die klassische Gewaltenteilung, wie noch beim C-64 gegeben, wo der VIC für die Grafik, der SID für den Sound und der CIA für die Eingabe/Ausgabe zuständig ist, ist beim C 16/116 und Plus/4 aufgehoben. Der TED vereint (fast) alle diese Aufgaben in sich. Diese Artikelserie stellt dieses Geschenk des Himmels nun mit allen seinen Möglichkeiten ausführlich vor, indem sie sich mit allen wichtigen TED-Registern intensiv beschäftigt. Eine Ausnahme bildet allerdings die Eingabe/Ausgabe-Programmierung, die nicht Gegenstand dieses Kurses sein soll (vielleicht ein andermal). Der Kurs richtet sich sowohl an die BASIC- als auch an die Maschinenprogrammierer und behandelt schwerpunktmäßig die Grafikprogrammierung des TED. Die folgende Inhaltsübersicht dient dazu, Ihnen einen Überblick über alle Themen des Kurses zu geben:

Inhaltsübersicht

1. Maskiertes POKEn
2. Umschalten des CPU-Zugriffes zwischen RAM und ROM
3. Die Grafikmöglichkeiten des TED
   I. Der Bitmap- und der Multicolor-Modus (Listing 1)
   II. der Extended Color-Modus (Listing 2)
   III. Ein selbsterstellter Zeichensatz (Listing 3)
4. Zeitvorteile durch Abschalten des Bildschirmes
5. Das Abfrageregister für Tastatur und Joystick
6. Die Interruptmöglichkeiten des TED
   I. Das IMR und das IRR
   II. Die Timer des TED (Listing 4)
   III. Der Rasterzeilen-IRQ (Listing5)
7. Softscrolling (Listing 6)
8. Umschalten zwischen PAL- und NTSC-Norm
9. Der Hardware-Cursor (Listing 7)
10. Die Soundmöglichkeiten des TED (Listing 8)

Maskiertes POKEn

Bekanntlich besteht der Speicher eines Computers aus einzelnen Bytes, von denen jedes einzelne wiederum aus acht Bits (numeriert von 0 bis 7) besteht. Ein Bit kennt zwei Zustände:
1 = gesetzt und 0 = ungesetzt.
Nun kann es vorkommen - und das wird im Laufe dieses Artikels mehrfach passieren -, daß man ein Bit oder auch mehrere Bits setzen oder löschen möchte bzw. muß, ohne jedoch den Zustand der übrigen Bits dieses Bytes in irgendeiner Weise zu verändern. Nötig wird das hauptsächlich bei Registern, wie sie in diesem Kurs ja beschrieben werden sollen, wo die einzelnen Bits - teilweise völlig unabhängig voneinander - bestimmte Steuerfunktioen erfüllen. Für dieses sogenannte "maskierte POKEn" kann man folgende Formel aufstellen:

Will man beispielsweise Bit n (n aus dem Bereich von 0 bis 7) des Registers XXXX setzen, ist von BASIC aus folgender Befehl notwendig:

```
POKE XXXX,
PEEK(XXXX) OR (2↑n)
```

Will man dasselbe Bit löschen, ist folgender Befehl einzugeben:

```
POKE XXXX,
PEEK(XXXX) AND
(255-2↑n)
```

Es empfiehlt sich natürlich die Terme hinter den logischen Verknüpfungen in ausgerechneter Form anzugeben, da das die Befehlsausführung erheblich beschleunigt. Will man mehrere Bits gleichzeitig löschen oder setzen, so sind die Terme für die einzelnen Bits zu addieren. Ein Beispiel:

```
POKE XXXX,
PEEK(XXXX) OR
(216+215+211)
```

setzt die Bits 6, 5 und 1 im Register XXXX auf 1. Zum Löschen:

POKE XXXX,
PEEK(XXXX) AND
(255-(216+215+211))

löscht dieselben Bits in dem besagten Register, ohne den Zustand der übrigen Bits zu beeinflussen. Wenn in einem Register bestimmte Bits gelöscht und andere wiederum gesetzt werden müssen, so ist natürlich auch eine Verknüpfung beider Befehle möglich. Folgender Befehl löscht Bit 0 und setzt Bit 1 im Register XXXX:

POKE XXXX,
(PEEK(XXXX) AND
(255-210)) OR (211)

In diesem Zusammenhang noch ein Hinweis zu den im nachfolgenden Text verwendeten Zahlenformaten. Um sowohl BASIC- als auch Maschinenprogrammierern gerecht zu werden, werden Zahlen bzw. Adressen grundsätzlich sowohl dezimal als auch hexadezimal - teilweise sogar binär - dargestellt. Dezimale Zahlenangaben haben deshalb entweder keine besondere Kennzeichnung oder sind mit "dez." gekennzeichnet, hexadezimale Zahlen mit "#$" und "hex." bzw. hexadezimale Adressen mit "$" und binäre Zahlen mit "#%" und "bin.".

Umschalten des CPU-Zugriffs zwischen ROM und RAM

Wie sicherlich viele unter den Lesern wissen, macht der C16 zur Verwaltung seines Speichers von der Technik des Bank-Switchings Gebrauch. Das heißt nichts weiter, als daß beim C16 "ständig" zwischen dem Festwertspeicher, ROM, und dem Anwenderspeicher, RAM, hin- und hergeschaltet wird und so die CPU Daten sowohl aus dem ROM als auch aus dem RAM lesen kann. Betrachtet man jedoch den Speicher des C16, so wirkt diese relativ aufwenige Technik der Speicherverwaltung wie die reinste Verschwendung: um 16 KByte RAM- und 32 KByte ROM-Speicher zu verwalten, bedarf es bei einer 8-Bit-CPU, wie sie mit dem 7501 im C16 integriert ist, wirklich keines Bank- Switchings. Der 7501 kann völlig ohne Tricks von Haus aus 64 KByte Speicher adressieren. Des Rätsels Lösung findet man leicht, angesichts der zahlreichen Speichererweiterungen, die seit einiger Zeit für den C16 angeboten werden. Ganz unkompliziert läßt sich der C16 nämlich auf 64 KByte hochrüsten. Der C16 ist schon ab Werk auf einen solchen 64 KByte-Betrieb vorbereitet. Es müssen praktisch nur zwei Speicherbausteine ausgewechselt werden, und schon haben Sie aus dem C16 eine 64 KByte-Maschine gemacht. Um Mißverständnissen vorzubeugen: durch Aufrüsten des Speichers wird der C16 nicht kompatibel zu "seinem großen Bruder C64"! Warum Commodore diesen Eingriff nicht gleich beim Zusammenbau des C 16 vornimmt, ist noch nicht ganz geklärt: entweder wollte man Verwechselungen mit dem C64 vermeiden oder aber dem Plus/4 nicht das Wasser abgraben; denn für den Plus/4 gilt folgende einfache Formel: Plus/4 = C16 + 64 KByte RAM + Userport + eingebaute Software.

Bei einem auf 64 KByte RAM hochgerüsteten C16 wird Sinn und Zweck des Bank-Switching schnell klar: der Prozessor des C 16 kann unmöglich 96 KByte Speicher (64 KByte RAM + 32 KByte ROM) auf einmal adressieren. Man bedient sich daher eines Tricks und greift nacheinander auf das RAM und das ROM zu. Per Bank-Switching schaltet man also zwischen einer ROM- und einer RAM-Bank um.

Um dem Anwender zu ermöglichen, Daten sowohl aus dem RAM als auch aus dem ROM zu lesen, existieren im TED zwei Schaltregister. Es sind dies die Register 62 (dez. 65342 / hex. $FF3E) und 63 (dez. 65343 / hex. $FF3F). Bei beiden Registern handelt es sich um Nur-Schreib-Speicher, das heißt, daß das Auslesen dieser Speicherzellen keine sinnvollen Werte liefert. Erst das Hineinschreiben eines (beliebigen) Wertes zeigt Wirkung: schreibt man in das Register 62, wird die ROM-Bank aktiviert, so daß nachfolgende Leseoperationen immer den Inhalt des ROM liefern; entsprechend wird durch das Beschreiben des Registers 63 die RAM-Bank selektiert. Bevor Sie nun versuchen, in diesen Registern herumzupoken, bedenken Sie folgendes: das gesamte Betriebssystem des C16 befindet sich im ROM. Deshalb wird beim Einschalten automatisch die ROM-Bank aktiviert. Wenn Sie nun von Basic aus in das Register 63 schreiben (per POKE-Befehl), also RAM selektieren, wird das ROM und damit das gesamte Betriebssystem sofort ausgeblendet. Und ohne die Steuerung durch das Betriebssystem muß der C16 unweigerlich abstürzen. Ein Umschalten zwischen ROM und RAM kann also nur innerhalb einer Maschinenroutine sinnvoll sein, solange diese nicht auf ROM-Betriebssystemroutinen zugreift.
Für den Ablauf eines eigenen Maschinenprogrammes ist das Betriebssystem nämlich nicht notwendig.

Die Grafikmöglichkeiten des TED

Neben dem nach dem Einschalten des Computers vorliegenden Normalmodus verfügt der TED-Chip noch über drei weitere Grafikmodi, die im folgenden beschrieben werden sollen. Es handelt sich dabei im einzelnen um den Extended Color-Modus, den einfachen Bitmap-Modus und den Multicolor-Modus.

Der Bitmap- und der Multicolor-Modus

Die beiden letzteren lassen sich dabei auf die Ihnen mit Sicheheit bekannte Weise über den Graphic-Befehl ansprechen. Diese beiden Modi werden durch Setzen eines Bits im TED-Register 6 bzw. 7 (Adressen $ FF06, dez. 65286 - und $ FF07, dez. 65287) eingeschaltet (genau das macht auch der Graphic-Befehl). Beim Bitmap-Modus ist das Bit 5 des TED-Registers 6, beim Multicolor-Modus zusätzlich das Bit 4 des TED-Registers 7 zuständig, d.h. beim Multicolor-Modus müssen auf jeden Fall beide Bits gesetzt sein. Der Graphic-Befehl schaltet aber nicht nur die entsprechenden Grafikmodi ein, sondern legt auch die Anfangsadresse des Bitmap-RAMs fest.

Dies geschieht über die Bits 3 und 5 des TED-Registers 18 (Adr. $ FF12, dez. 65298), die die drei höchstwertigsten Bits, d. h. die Bits 13 und 15, der Adresse des Bitmap-RAMs enthalten. Die Adress-Bits 0 bis 12 haben also den unveränderbaren Wert 0. Durch den Graphic-Befehl wird in diese drei höchstwertigsten Bits der Wert #% 001 (binär) geschrieben und damit die Adresse für das Bitmap-RAM mit $ 2000 (dez. 8192) festgelegt. Eine andere Möglichkeit existiert in der Grundversion des C 16/116 auch gar nicht, da das Setzen eines der beiden höchstwertigsten Adress-Bits auf jeden Fall eine Bitmap-RAM-Adresse ergibt, die größer als $ 3FFF (dez. 16383) ist und somit außerhalb des zur Verfügung stehenden RAM-Speicherbereiches liegt. Die Bitmap-RAM-Adresse für $ 0000 festzulegen (d. h. alle drei höchstwertigsten Bits zu löschen), kommt natürlich überhaupt nicht in Frage, da die Zero-Page (Speicherstellen $ 0000 bis $ 00FF) vom Betriebssystem und Basic-Interpreter genutzt werden und vor dem Bitmap-RAM noch das Farbram und die Luminanztabelle liegen müssen.

Das Bitmap-RAM läßt sich also nur bei einem erweiterten C 16=116 bzw. beim Plus/4 frei im Speicher verschieben. Zur Orientierung dient folgende kurze Tabelle, die zu jeder Konstellation der drei höchstwertigsten Adress-Bits die entspechende Position des Bitmap-Rams im Speicher angibt:

Bits 3 - 5 des TED-Reg. 18 (Adresse Bits 13 - 15)	Die Adresse des Bitmap-RAMs
000	$ 0000 (dez. 0) - nicht möglich
001	$ 2000 (dez. 8192) - normal
010	$ 4000 (dez. 16384)
011	$ 6000 (dez. 24576)
100	$ 8000 (dez. 32768)
101	$ A000 (dez. 40960)
110	$ C000 (dez. 49152)
111	$ E000 (dez. 57344)

Tabelle 1

Normalerweise bezieht der TED alle seine Grafikinformationen aus dem ROM. Im Falle einer Hires-Grafik liegen diese aber im RAM, so daß dem TED noch mitgeteilt werden muß, daß er seine Grafikinformationen aus dem RAM beziehen soll. Dies geht

über ein spezielles Umschaltebit, das sich RAM/ROM Bank-Bit nennt. Es handelt sich dabei um Bit 2 im TED-Register 18 (Adr. $ FF12, dez. 65298). Ist es gesetzt, so wird vom TED das ROM, ist es gelöscht, so das RAM angesprochen. Damit ist es auch möglich beim erweiterten C 16/116 bzw. Plus/4 eine Grafik unter das ROM zu legen. Das Umschalten von ROM auf RAM erledigt der Graphic-Befehl selbstverständlich auch. Wollen Sie ein unter das ROM gelegtes Bitmap-RAM von Maschinensprache aus bearbeiten, so müssen Sie noch durch Setzen eines Wertes im TED-Register 63 (siehe vorhergehenden Artikelabschnitt) dafür sorgen, daß auch die CPU auf das RAM zugreift. Die Grafikroutinen des Basic-Interpreters können Sie eh nur verwenden, wenn Sie diese ins RAM kopieren und dort in der Unterroutine "Bytezeiger setzen" (Adr. $ C291) die Basicadresse des Bitmap-RAMs in $ C2A9 modifizieren. Als das zum Bitmap-RAM gehörende Farbram fungiert immer das Videoram. Damit aber nicht durch das Arbeiten im Bitmap- bzw. Multicolor-Modus die Informationen im "normalen" Videoram (womit das Videoram ab Adresse $ 0C00 -dez. 3072-, wie im Einschaltzustand gegeben, gemeint ist) zerstört werden, muß auch das Videoram zunächst an eine neue Stelle im Speicher verschoben und nach dem Abschalten des Bitmap-Modus' wieder an die alte Adresse zurückgeschoben werden. Ähnlich wie das Bitmap-RAM läßt sich also auch das Videoram verschieben, und zwar über die Bits 3 bis 7 (Adress-Bits 11 bis 15 des Videorams) im TED-Register 20 mit der Adresse $ FF14 (dez. 65300). Aus den gleichen auch schon beim Bitmap-RAM angegebenen Gründen sind natürlich nicht alle Adressen möglich. Durch den Graphic-Befehl wird das Videoram ab Adresse $ 1000 (dez. 7168), also direkt vor dem Bitmap-RAM installiert. Auch hier wieder eine Übersicht der Adressen des Videorams in Abhängigkeit von dem Wert der fünf höchstwertigsten Adress-Bits:

Bits 3-7 des Videorams	Adresse des TED-Reg. 20
00000	$ 0400 (dez. 1024)
00001	$ 0C00 (dez. 3072)
00010	$ 1400 (dez. 5120)
00011	$ 1C00 (dez. 7168)
00100	$ 2400 (dez. 9216)
00101	$ 2C00 (dez. 11264)
00110	$ 3400 (dez. 13312)
00111	$ 3C00 (dez. 15360)
01000	$ 4400 (dez. 17408)
01001	$ 4C00 (dez. 19456)
01010	$ 5400 (dez. 21504)
01011	$ 5C00 (dez. 23552)
01100	$ 6400 (dez. 25600)
01101	$ 6C00 (dez. 27648)
01110	$ 7400 (dez. 29696)
01111	$ 7C00 (dez. 31744)
10000	$ 8400 (dez. 33792)
10001	$ 8C00 (dez. 35840)
10010	$ 9400 (dez. 37888)
10011	$ 9C00 (dez. 39936)
10100	$ A400 (dez. 41984)
10101	$ AC00 (dez. 44032)
10110	$ B400 (dez. 46080)
10111	$ BC00 (dez. 48128)
11000	$ C400 (dez. 50176)
11001	$ CC00 (dez. 52224)
11010	$ D400 (dez. 54272)
11011	$ DC00 (dez. 56320)
11100	$ E400 (dez. 58368)
11101	$ EC00 (dez. 60416)
11110	$ F400 (dez. 62464)
11111	$ FC00 (dez. 64512)

Tabelle 2

Da Bit 10 der Videoram-Adresse also immer gesetzt ist (die Bits 0 bis 9 sind immer 0), wäre die erstmögliche Adresse theoretisch $ 0400.

Diese Adresse kommt aber aus den auch schon beim Bitmap-RAM genannten Gründen nicht in Frage, so daß die im Einschaltzustand gegebene Adresse $ 0C00 die erstmögliche Position für das Videoram darstellt. Nun soll noch kurz geklärt werden, warum eine Hires-Grafik überhaupt so viel Speicherplatz, nämlich etwa 10 Kilobyte, in Anspruch nimmt. Wie Sie sicherlich wissen, hat der C 16/116 bzw. Plus/4 eine Grafikauflösung von 320 x 200 = 64000 Punkten. Die Zustände von acht Punkten (gesetzt oder nicht gesetzt) lassen sich jeweils in einem Byte zusammenfassen. Damit werden 64000 / 8 Bytes = 8000 Bytes allein für das Bitmap-RAM benötigt. Weitere 1000 Bytes (25 Zeilen X 40 Spalten) werden für das Farbram gebraucht und ebenfalls noch einmal 1000 Bytes für die Luminanztabelle, die beim Einschalten des Hires-Modus ab Adresse $ 1800 (dez. 6144), also direkt vor dem Farbram liegt (das ist automatisch immer der Fall). Beim Multicolor-Modus wird die Auflösung in X-Richtung auf 160 Punkte verkürzt, weil jetzt jeweils zwei Bits für einen Grafikpunkt nötig sind, da ja auch die Farbinformation enthalten sein muß. Mit zwei Bits lassen sich 2↑2 = 4 Zustände darstellen (00, 01, 10, 11) ,so daß ein Punkt vier verschiedene Farben (allerdings die Hintergrundfarbe eingeschlossen) innerhalb eines Zeichenbereiches (8 x 8 = 64 Punkte) annehmen kann. Die Behandlung des Bitmap- und des Multicolor-Modus von Basic aus ist relativ einfach und komfortabel. Wie Sie auch von Maschinensprache her unter Ausnutzung der gegebenen Grafikroutinen im Basic-ROM mit Hires-Grafiken arbeiten können, soll anhand von Listing 1 gezeigt werden. In diese Listing wird die Handhabung aller wichtigen Grafikroutinen demonstriert, d. h. es werden die an die Grafikroutinen zu übergebenden Parameter und die Einsprungadressen in mehreren hintereinandergehängten Beispielen vorgeführt.

Alles weitere ist der Kommentierung des im Basic-Loader integrierten Assemblerlistings zu entnehmen.

Der Extended Color-Modus

Eine interessante Fähigkeit des TED ist der Extended Color-Modus. In ihm läßt sich jedes Zeichen auf dem Bildschirm in einer eigenen Hintergrundfarbe darstellen. Allerdings stehen in diesem Modus nur noch sechs Bits für die Zeichendefinition zur Verfügung, da die beiden höchstwertigsten Bits zur Festlegung der Hintergrundfarbe herangezogen werden. Mit sechs Bits lassen sich aber nur noch 216 = 64 Zeichen definieren.

Da es sich bei diesen sechs Bits um die sechs niederwertigsten Bits handelt, sind es also die ersten 64 in der Bildschirmcode-Tabelle (s. HB. S. 216f) angebenen Zeichen. Zwei Bits ergeben vier mögliche Hintergrundfarben, die in den TED-Registern 21 (Adr. $ FF15, dez. 65301) bis 24 (Adr. $ FF18, dez. 65304) festgelegt sind.

In diesem Zusammenhang soll auch das TED-Register 25 (Adr. $ FF19, dez. 65305), das die Rahmenfolge festlegt, nicht unerwähnt bleiben. Die niederwertigsten vier Bits, d. h. die Bits 0 bis 3, in diesen fünf Registern legen den Farbwert fest, de Bits 4 bis 6 die Luminanz und Bit 7 ist ungenutzt (im Farbram ist Bit 7 dann gesetzt, wenn das Zeichen blinkend erscheinen soll).

Über den Color-Befehl lassen sich unglücklicherweise nur die ersten beiden und das letzte der erwähnten fünf Register ansprechen. Ist der erste Parameter des Color-Befehles 0, so wird der folgende Farb- und Luminanzwert in das TED-Register 21, ist er 3, dann in das TED-Register 22 und bei 4 in das TED-Register 25 geschrieben.

In die Register 23 und 24 müssen Farb- und Luminanzwert also mit POKE gesetzt werden. Dabei ist zu beachten, daß vom gewünschten Farbwert 1 abgezogen und der Luminanzwert mit 16 multipliziert (da ja die höchstwertigsten Bits 4 bis 6 diesen Wert festlegen) hinzuaddiert werden muß.

Da, wie bereits erwähnt, im Extended Color-Modus nur noch 64 verschiedene Zeichen zur Verfügung stehen, entsprechen die Bildschirmcodes 0 bis 63 diesen ersten 64 Zeichen mit der Hintergrundfarbe im TED-Register 21, die Bildschirmcodes 64 bis 127 entsprechend den ersten 64 Zeichen mit der Hintergrundfarbe im

```
100 REM ****************************
105 REM *                          *
110 REM * NUTZUNG DER GRAFIKROUTINEN *
115 REM * VON MASCHINENSPRACHE AUS   *
120 REM *                          *
125 REM * 1986 BY D.D.+W. ..
130 REM *                          *
135 REM ****************************
140 :
145 :
150 PRINT"■ACHTUNG: BASIC-LOADER LOESCHT SICH"
155 PRINT"SELBST!":PRINT"WEITER MIT BELIEBIGER TASTE."
160 GETKEYT$:PRINT"■BITTE KURZ WARTEN!"
165 FORA=1630TO1897:READD:P=P+D:POKEA,D:NEXT
170 IFP<>28727THENPRINT"■FEHLER IN DEN DATA-ZEILEN!":END
175 FORT=1TO20:PRINT"■":NEXT
180 PRINT"STARTEN MIT SYS 1630.":NEW
185 :
190 :
195 REM  ALLE PARAMETER BEZIEHEN SICH AUF DEN MULTICOLOR-MODUS, D.H. DIE
200 REM  X-WERTE SIND IMMER KLEINER ALS 160.
205 :
210 :
215 * FARBPARAMMETER SETZEN UND GRAFIK INITIALISIEREN (COLOR & GRAPHIC) *
220 :
225 DATA169,66      :       LDA #$42       " FARBE: 2 (ROT)  / LUMINANZ: 4
230 DATA133,134     :       STA $86        " IN FARBREGISTER #1 SCHREIBEN
235 DATA169,83      :       LDA #$53       " FARBE: 3 (CYN)  / LUMINANZ: 5
240 DATA133,133     :       STA $85        " IN FARBREGISTER #2 SCHREIBEN
245 DATA169,100     :       LDA #$64       " FARBE: 4 (PUR)  / LUMINANZ: 6
250 DATA141,22,255  :       STA $FF16      " IN FARBREGISTER #3 SCHREIBEN
255 DATA162,4       :       LDX #$04       " GRAFIKMODUS 4 (MULTICOLOR+TEXT)
260 DATA32,216,197  :       JSR $C5D8      " ==> GRAFIK INITIALISIEREN
265 :
270 * SCNCLR (SCREEN CLEAR) *
275 :
280 DATA32,127,197  :       JSR $C57F      " ==> GRAFIKBILDSCHIRM LOESCHEN
285 :
290 * LINIE BZW. PUNKT ZEICHNEN (DRAW) *
295 :
300 DATA169,1       :       LDA #$01       " FARBREGISTER #1 ALS AKTUELLE
305 DATA133,132     :       STA $84        " ZEICHENFARBE SELEKTIEREN
310 DATA169,0       :       LDA #$00       "
315 DATA141,173,2   :       STA $02AD      " STARTKOORDINATE: X-LO-BYTE
320 DATA141,174,2   :       STA $02AE      "                  X-HI-BYTE
325 DATA141,175,2   :       STA $02AF      "                  Y-LO-BYTE
330 DATA141,176,2   :       STA $02B0      "                  Y-HI-BYTE
335 DATA169,48      :       LDA #$30       "
340 DATA162,0       :       LDX #$00       "
345 DATA141,177,2   :       STA $02B1      " ZIELKOORDINATE : X-LO-BYTE
350 DATA142,178,2   :       STX $02B2      "                  X-HI-BYTE
355 DATA169,32      :       LDA #$20       "
360 DATA141,179,2   :       STA $02B3      "                  Y-LO-BYTE
365 DATA142,180,2   :       STX $02B4      "                  Y-HI-BYTE
370 DATA32,218,192  :       JSR $C0DA      " START- UND ZIELKOOR. VERBINDEN
375 :
380 * KREIS ZEICHNEN (CIRCLE) *
385 :
390 DATA169,2       :       LDA #$02       " FARBRREGISTER #2 ALS AKTUELLE
395 DATA133,132     :       STA $84        " ZEICHENFARBE SELEKTIEREN
400 DATA169,80      :       LDA #$50       "
405 DATA162,0       :       LDX #$00       "
410 DATA141,204,2   :       STA $02CC      " MITTELPUNKTKOOR.: X-LO-BYTE
415 DATA142,205,2   :       STX $02CD      "                   X-HI-BYTE
420 DATA169,48      :       LDA #$30       "
425 DATA141,206,2   :       STA $02CE      "                   Y-LO-BYTE
430 DATA142,207,2   :       STX $02CF      "                   Y-HI-BYTE
435 DATA169,16      :       LDA #$10       "
440 DATA141,208,2   :       STA $02D0      " X-RADIUS: LO-BYTE
445 DATA142,209,2   :       STX $02D1      "           HI-BYTE
450 DATA169,32      :       LDA #$20       "
455 DATA141,210,2   :       STA $02D2      " Y-RADIUS: LO-BYTE
460 DATA142,211,2   :       STX $02D3      "           HI-BYTE
465 DATA142,216,2   :       STX $02DB      " STARTWINKEL : LO-BYTE (=0)
470 DATA142,217,2   :       STX $02D9      "               HI-BYTE (=0)
475 DATA142,218,2   :       STX $02DA      " ENDWINKEL   : LO-BYTE (=0)
480 DATA142,219,2   :       STX $02DB      "               HI-BYTE (=0)
485 DATA142,212,2   :       STX $02D4      " DREHWINKEL  : LO-BYTE (=0)
490 DATA142,213,2   :       STX $02D5      "               HI-BYTE (=0)
495 DATA169,2       :       LDA #$02       " SCHRITTWEITE (BESTIMMT DIE LAENGE
500 DATA133,233     :       STA $E9        " DER KREISSEGMENTE) AUF 2 SETZEN
505 DATA32,101,192  :       JSR $C065      " KREIS ZEICHNEN
```

Listing 1, 1. Teil

TED-Register 22 usw.. Es gelten also folgende Zuordnungen:

(siehe Tabelle 3)

Einschalten läßt sich der Extended Color-Modus durch Setzen von Bit 6 im TED-Register 6 (Adr. $ FF06, dez. 65286). Listing 2 soll den Umgang mit diesem Grafikmodus demonstrieren.

Ein selbsterstellter Zeichensatz

Der TED-Chip bietet die Möglichkeit, sich seinen eigenen Zeichensatz zu erstellen. Diese wird von vielen professionellen Spielen genutzt, da das Arbeiten mit undefinierten Zeichen große Geschwindigkeitsvorteile gegenüber dem Hires-Modus einbringt. Normalerweise bezieht der TED seine Zeicheninformationen aus dem Character-ROM, das von Adresse $ D000 (dez. 53248) bis Adresse $ D7FF (dez. 55295) liegt. Im Character-ROM sind die Zeichendefinitionen für 128 ASCII-Zeichen jeweils für den Großschrift/Grafikzeichen- und den Kleinschrift/Großschrift-Modus gespeichert, und zwar in der Reihenfolge, wie sie in der Bildschirmcode-Tabelle auf S. 216/217 im Handbuch angegeben ist. Die dazu entsprechenden reversen Zeichen mit den Bildschirmcodes 128 bis 255 werden handwaremäßig erzeugt. Jedes Zeichen im Character-ROM beansprucht acht Bytes, da ein Zeichen aus 8 x 8 = 64 Punkten zusammengesetzt ist, wovon sich jeweils acht Punkte in einem Byte (= 8 Bits) zusammenfassen lassen. Jedes Byte steht dabei für eine Punktzeile des Zeichens, wobei ein gesetztes Bit auch einem gesetzten Punkt entspricht. Wollen Sie also ein eigenes Zeichen definieren, so müssen Sie das Zeichen zunächst in einem 8 x 8-Raster erstellen und dann für jede Zeile den aus den gesetzten Punkten (= Bits) resultierenden Byte-Wert errechnen. Das soll am Beispiel des Buchstaben A, der in der Bildschirmcode-Tabelle das zweite Zeichen und im Character-ROM demnach ab Adr. $ D008 (dez. 53256) abgelegt ist, gezeigt werden. Das Punkteraster für den Buchstaben A sähe also wie folgt aus (ein Stern (*) markiert immer einen gesetzten Punkt):

(siehe Tabelle 4)

```
510 :
515 * VIERECK ZEICHNEN (BOX) *
520 :
525 DATA169,112     :     LDA #$70      "
530 DATA162,0       :     LDX #$00      "
535 DATA141,204,2   :     STA $02CC     " LINKE OBERE ECKE  : X-LO-BYTE
540 DATA142,205,2   :     STX $02CD     "                     X-HI-BYTE
545 DATA169,80      :     LDA #$50      "
550 DATA141,206,2   :     STA $02CE     "                     Y-LO-BYTE
555 DATA142,207,2   :     STX $02CF     "                     Y-HI-BYTE
560 DATA169,144     :     LDA #$90      "
565 DATA141,216,2   :     STA $02D8     " RECHTE UNTERE ECKE: X-LO-BYTE
570 DATA142,217,2   :     STX $02D9     "                     X-HI-BYTE
575 DATA169,112     :     LDA #$70      "
580 DATA141,218,2   :     STA $02DA     "                     Y-LO-BYTE
585 DATA142,219,2   :     STX $02DB     "                     Y-HI-BYTE
590 DATA141,208,2   :     STA $02D0     " DREHWINKEL : LO-BYTE
595 DATA142,209,2   :     STX $02D1     "              HI-BYTE
600 DATA162,1       :     LDX #$01      " FLAG FUER FLAECHE AUSFUELLEN SETZEN
605 DATA32,2,187    :     JSR $BB02     " VIERECK ZEICHNEN
610 :
615 * FLAECHE AUSFUELLEN (PAINT) *
620 :
625 DATA169,3       :     LDA #$03      " FARBREGISTER #3 ALS AKTUELLE
630 DATA133,132     :     STA $84       " ZEICHENFARBE SELEKTIEREN
635 DATA169,80      :     LDA #$50      "
640 DATA162,0       :     LDX #$00      "
645 DATA141,173,2   :     STA $02AD     " STARTWERT : X-LO-BYTE
650 DATA142,174,2   :     STX $02AE     "             X-HI-BYTE
655 DATA169,48      :     LDA #$30      "
660 DATA141,175,2   :     STA $02AF     "             Y-LO-BYTE
665 DATA142,176,2   :     STX $02B0     "             Y-HI-BYTE
670 DATA162,1       :     LDX #$01      " FUELLMODUS AUF EINS SETZEN
675 DATA32,230,184  :     JSR $BBE6     " FLAECHE AUSFUELLEN
680 :
685 * TEXT IN GRAFIKBILDSCHIRM SCHREIBEN (CHAR) *
690 :
695 DATA169,1       :     LDA #$01      " FARBREGISTER #1 ALS AKTUELLE
700 DATA133,132     :     STA $84       " ZEICHENFARBE SELEKTIEREN
705 DATA141,218,2   :     STA $02DA     " AKTUELLE SPALTE = 1
710 DATA169,18      :     LDA #$12      "
715 DATA141,219,2   :     STA $02DB     " AKTUELLE ZEILE  = 18
720 DATA169,34      :     LDA #$22      "
725 DATA141,234,2   :     STA $02EA     " STRINGLAENGE    = 34
730 DATA169,66      :     LDA #$42      "
735 DATA162,07      :     LDX #$07      "
740 DATA133,34      :     STA $22       " STRINGADRESSE : LO-BYTE
745 DATA134,35      :     STX $23       "                 HI-BYTE
750 DATA169,128     :     LDA #$80      "
755 DATA141,185,2   :     STA $02B9     " REVERS-FLAG SETZEN
760 DATA76,9,186    :     JMP $BA09     " STRING AUSGEBEN
765 :
770 * TEXTTABELLE FUER CHAR *
775 :
780 DATA77,85,76,84,73,67,79,76,79,82,45,68,69,77,79,32,68,69,82,32,71,82
785 DATA65,70,73,75,82,79,85,84,73,78,69,78,72,69,32,65,85,83          READY.
```

Listing 1, 2. Teil

```
Adresse Bit 76543210
$ D008    ...**...  = # %00011000 = #$18 = dez. 24
$ D009    ..****..  = # %00111100 = #$3C = dez. 60
$ D00A    .**..**.  = # %01100110 = #$66 = dez. 102
$ D00B    .******.  = # %01111110 = #$7E = dez. 126
$ D00C    .**..**.  = # %01100110 = #$66 = dez. 102
$ D00D    .**..**.  = # %01100110 = #$66 = dez. 102
$ D00E    .**..**.  = # %01100110 = #$66 = dez. 102
$ D00F    ........  = # %00000000 = #$00 = dez. 0
```

Tabelle 3

entsprechende Zeichen	Adresse des zugehörigen TED-Registers	Wert der beiden Bildschirmcodes	entsprechende höchstwertigsten Bits
0 - 63	$ FF15 (dez.65301)	00	0 - 63
0 - 63	$ FF16 (dez. 65302)	01	4 - 127
0 - 63	$ FF17 (dez. 65303)	10	128 - 191
0 - 63	$ FF18 (dez. 65304)	11	192 - 255

Tabelle 4

Nun ist das ROM aber leider (bzw. zum Glück) unveränderbar; folglich muß ein Zeichensatz mit selbstdefinierten Zeichen im RAM liegen. Dazu muß zunächst die Adresse des Zeichensatzes geändert werden. Dies ist über die Bits 2 bis 7 des TED-Registers 19 (Adr. $ FF13, dez. 65299), bei denen es sich um die Adress-Bits 10 bis 15 des Zeichensatzes handelt, möglich. Wie schon beim Bitmap-RAM haben auch hier wieder die niederwertigsten Adress-Bits, d. h. in diesem Fall die Bits 0 bis 9, den festen Wert 0. Auf eine Tabelle mit den verschiedenen Bitkonfigurationen und den zugehörigen Adressen des Zeichensatzes wurde hier wegen der Vielzahl der Möglichkeiten (nämlich genau 216 = 64) verzichtet. Vermutlich stellt es Sie aber auch nicht vor allzu große Probleme, die zur gewünschten Zeichensatzadresse benötigte Bitkonfiguration selbst herauszufinden. Ist die Zeichensatzadresse geändert, so muß dem TED noch mitgeteilt werden, daß er die Zeichendefinitionen nicht mehr aus dem ROM, sondern aus dem RAM beziehen soll. Dazu wird das schon erwähnte RAM/ROM Bank-Bit (Bit2) im TED-Register 18 gelöscht. Damit wäre also der neue Zeichensatz eingeschaltet. Eine weitere interessante Fähigkeit des TED darf in diesem Zusammenhang nicht unerwähnt bleiben. Normalerweise lassen sich auf diese Art nur 128 Zeichen selbst definieren, da, wie bereits gesagt, die reversen Zeichen hardwaremäßig erzeugt werden. D. h., wenn Bit 7 im Bildschirmcode eines Zeichens gesetzt ist, so wird dieses automatisch vom TED als reverses Gegenstück desselben Zeichens mit nicht gesetztem Bit 7 interpretiert und dementsprechend dargestellt. Dieses Verhalten des TEDs läßt sich aber durch Setzen des Bits 7 im TED-Register 7 (Adr. $ FF07, dez. 65287) ändern. Dieses Bit entscheidet nämlich darüber, ob die zweiten 128 Zeichen hardware- oder softwaremäßig erzeugt werden. Ist es also gesetzt, so werden auch die Zeichen mit den Bildschirmcodes 128 bis 255 softwaremäßig erzeugt und können nach Belieben neu definiert werden. Allerdings steigt dann natürlich der Platzbedarf für den Zeichensatz um das Doppelte. Will man einen eigenen Zeichensatz erstellen, aber noch Zeichen aus dem normalen Zeichensatz übernehmen, so ist es am günstigsten, den gesamten Zeichensatz aus dem Character-ROM ins RAM zu kopieren. Das Beispielprogramm Listing 3 erstellt einen Zeichensatz mit den deutschen Umlauten und ß (sz). Die neuen Zeichen erhalten die Bildschirmcodes 65 (Ä), 79 (Ö),

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

100 REM ************************                    <27>
105 REM *                      *                    <202>
110 REM * DEMO ZUM              *                    <3>
115 REM * EXTENDED COLOR-MODUS *                    <69>
120 REM *                      *                    <217>
125 REM * 1986 BY D.D.+W.R.    *                    <210>
130 REM *                      *                    <227>
135 REM ************************                    <62>
140 :                                               <198>
145 :                                               <203>
150 REM FARBEN FESTLEGEN                            <27>
155 :                                               <213>
160 COLOR0,1,3:REM HINTERGRUNDFARBE SCHWARZ         <141>
165 COLOR1,3,3:REM ZEICHENFARBE ROT                 <76>
170 COLOR3,7,6:REM ZUSATZFARBE BLAU                 <202>
175 POKE65303,5+4*16:REM ZUSATZFARBE GRUEN          <149>
180 POKE65304,7+5*16:REM ZUSATZFARBE GELB           <125>
185 :                                               <243>
190 REM EXTENDED COLOR-MODUS EINSCHALTEN            <230>
195 :                                               <253>
200 POKE65286,PEEK(65286)OR64                       <102>
205 :                                               <7>
210 SCNCLR:REM BILDSCHIRM LOESCHEN                  <15>
215 :                                               <17>
220 REM ZEICHEN AUF DEN BILDSCHIRM SETZEN           <39>
225 :                                               <27>
230 FORP=3352TO3380:READZ:POKEP,Z:NEXTP             <96>
235 FORP=3432TO3687:POKEP,P-3432:NEXTP              <9>
240 DATA4,69,141,207,32,90,149,205,32,69,152,212
,5,78,132,197,4,96,131                              <28>
245 DATA207,12,79,146,237,13,79,132,213,19          <133>
250 :                                               <52>
255 PRINT"{BLUE SZ}U{SM SPACE}A{SU}S{SS}C{SH}A{S
L}T{SE}N{SPACE WHITE RVSON SB}E{SL}I{SE}B{SI}G{S
E RVSOFF SPACE RVSON}T{SA}S{ST}E{RVSOFF SPACE9 G
REEN SD}R{SU}E{SC}K{SE}N!"                          <163>
260 GETKEYT$:REM AUF TASTENDRUCK WARTEN             <57>
265 :                                               <68>
270 REM EXTENDED COLOR-MODUS AUSSCHALTEN            <94>
275 :                                               <78>
280 POKE65286,PEEK(65286)AND191                     <79>
  ENDE DES LISTINGS
```

Listing 2

83 (ß) und 85 (Ü), sind also über den dem neuen Zeichen zugrundeliegenden Buchstaben in Kombination mit Shift zu erreichen (Ä also z. B. über Shift A). Da das Kopieren des Zeichensatzes über PEEK- und POKE-Befehle recht lange dauert, übernimmt diese Aufgabe im Beispielprogramm eine kleine Maschinenroutine (Basiclader in den Zeilen 175 bis 190), die Sie auch in eigene Programme integrieren sollten. Falls Sie doch über PEEK und POKE kopieren wollen, so stellt sich noch ein kleines Problem. Der PEEK-Befehl liest nämlich normalerweise immer aus dem RAM. Beim Plus/4 oder dem erweiterten C 16/116, wo sich ja bekannterweise RAM und ROM ab Adresse $ 8000 (dez. 32768) überschneiden, ist das verständlich. Aber auch bei der Grundversion des C 16/116 kann das ROM über PEEK nicht ausgelesen werden, da ja theoretisch ein RAM darunterliegen könnte. Die Entscheidung darüber, ob das ROM oder das RAM angesprochen wird, wird in den im vorhergehenden Artikelabschnitt bereits erklärten TED-Registern 62 und 63 getroffen. Glücklicherweise haben jedoch die Schlitzohren von Commodore an den Anwender gedacht und die Routine, die u.a. beim PEEK-Befehl die Festlegung ROM/RAM trifft, doch tatsächlich ins RAM gelegt. D.h., ganz so selbstlos war man bei Commodore vielleicht doch nicht; es gibt nämlich gar keine andere Möglichkeit, da das ROM ja durch diese Routine ausgeschaltet wird. Sie liegt im Bereich von Adresse $ 0494 (dez. 1172) bis $ 04A1 (dez. 1185), wo sie von uns modifiziert werden kann. Dazu genügt der Befehl POKE 1177, 62, der vor dem Kopieren des Character-ROMs von Basic aus stehen muß. Danach muß die Routine mit POKE 1177, 63 wieder richtiggestellt werden. Denken Sie bei eigenen Programmen auch daran, daß der neue Zeichensatz (ZS), wenn Sie ihn nicht gerade über Maschinenprogram-

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

100 REM ************************                    <95>
105 REM *                      *                    <202>
110 REM * DEUTSCHER ZEICHENSATZ *                    <199>
115 REM *                      *                    <212>
120 REM * 1986 BY D.D.+W.R.    *                    <205>
125 REM *                      *                    <222>
130 REM ************************                    <125>
135 :                                               <193>
140 :                                               <198>
145 REM NEUEN ZEICHENSATZ VOR UEBERSCHREIBEN DUR
CH BASIC SCHUETZEN                                  <13>
150 :                                               <208>
155 POKE55,0:POKE56,56                              <109>
160 :                                               <218>
165 REM CHARACTER-ROM KOPIERROUTINE                 <117>
170 :                                               <228>
175 FORA=900TO930:READD:POKEA,D:NEXTA               <133>
180 SYS900:REM ROUTINE STARTEN                      <238>
185 DATA160,0,132,217,132,219,169,208,133,218,16
9,56,133,220,162,7,177,217                          <251>
190 DATA145,219,200,208,249,230,218,230,220,202,
16,242,96                                           <74>
195 :                                               <253>
200 REM NEUE ZEICHEN IMPLEMENTIEREN                 <199>
205 :                                               <7>
210 ADR=14336+8*65                                  <41>
215 FORA=ADRTOADR+7:READD:POKEA,D:NEXTA             <20>
220 DATA102:REM .**..**.                            <5>
225 DATA24 :REM ...**...                            <115>
230 DATA60 :REM ..****..                            <40>
235 DATA102:REM .**..**.                            <20>
240 DATA126:REM .******.                            <219>
245 DATA102:REM .**..**.                            <30>
250 DATA102:REM .**..**.                            <35>
255 DATA0  :REM ........                            <4>
260 :                                               <63>
265 ADR=14336+8*79                                  <166>
270 FORA=ADRTOADR+7:READD:POKEA,D:NEXTA             <76>
275 DATA102:REM .**..**.                            <61>
280 DATA60 :REM ..****..                            <91>
285 DATA102:REM .**..**.                            <71>
290 DATA102:REM .**..**.                            <76>
295 DATA102:REM .**..**.                            <81>
300 DATA102:REM .**..**.                            <86>
305 DATA60 :REM ..****..                            <116>
310 DATA0  :REM ........                            <60>
315 :                                               <118>
320 ADR=14336+8*85                                  <178>
325 FORA=ADRTOADR+7:READD:POKEA,D:NEXTA             <131>
330 DATA102:REM .**..**.                            <116>
335 DATA0  :REM ........                            <85>
340 DATA102:REM .**..**.                            <126>
345 DATA102:REM .**..**.                            <131>
350 DATA102:REM .**..**.                            <136>
355 DATA102:REM .**..**.                            <141>
360 DATA60 :REM ..****..                            <171>
365 DATA0  :REM ........                            <115>
370 :                                               <173>
375 ADR=14336+8*83                                  <205>
380 FORA=ADRTOADR+7:READD:POKEA,D:NEXTA             <186>
385 DATA120:REM .*****..                            <137>
390 DATA102:REM .**..**.                            <176>
395 DATA102:REM .**..**.                            <181>
400 DATA120:REM .*****..                            <152>
405 DATA102:REM .**..**.                            <191>
410 DATA102:REM .**..**.                            <196>
415 DATA108:REM .**.**..                            <233>
420 DATA96 :REM .**.....                            <91>
425 :                                               <228>
430 REM NEUEN ZEICHENSATZ EINSCHALTEN               <21>
435 :                                               <238>
440 POKE65299,(PEEK(65299)AND3)OR56:REM NEUE ZEI
CHENSATZ-ADRESSE FESTLEGEN                          <165>
445 POKE65298,PEEK(65298)AND251:REM ZEICHENSATZ
AKTIVIEREN                                          <163>
450 :                                               <253>
455 PRINT"A{SPACE SA SPACE}O{SPACE SO SPACE}U{SP
ACE SU SPACE}S{SPACE SS}":REM NEUE ZEICHEN          <133>
  ENDE DES LISTINGS
```

Listing 3

me nutzen, unbedingt mit POKE 55, LO-Byte der ZS-Adr.: POKE 56, Hi-Byte der ZS-Adr. vor Überschreiben durch Basic geschützt werden muß. Um einem Schockerlebnis vorzubeugen, muß noch angemerkt werden, daß das Verursachen einer Fehlermeldung bei eingeschaltetem eigenen Zeichensatz schwerwiegende Folgen nach sich ziehen kann. Ihr Computer wird Sie in diesem Fall nämlich mit einer wundervoll hin- und herwogenden Grafik beglücken, die jedoch die Freude am Weiterarbeiten merklich trüben kann. Dies ist darauf zurückzuführen, daß mit der Auslösung einer Fehlermeldung auch die Hires-Grafik auf jeden Fall abgeschaltet wird (unabhängig davon, ob sie überhaupt eingeschaltet war oder nicht). Mit dem Abschalten der Hires-Grafik wird aber auch das RAM/ROM Bank-Bit in TED-Register 18 gesetzt, was dazu führt, daß der TED nicht mehr das RAM, sondern einen nicht existenten Zeichensatz im ROM anspricht (außer der Zeichensatz im RAM besitzt die gleiche Adresse wie der im ROM). Das ist also der Grund, warum uns der TED bei einer Fehlmeldung mit diesem widerlichen Grafik-Wirrwarr überrascht. Mit dem Löschen des RAM/ROM Bank-Bits durch POKE 65298, PEEK (65298) AND 251 läßt sich dem Zauber ein schnelles Ende bereiten.

Selbsterstellte Zeichen im Hires-Modus

Bekanntlich lassen sich im Hires-Modus Zeichen nur über den CHAR-Befehl setzen. Anders allerdings als im Normalmodus wird im Hires-Modus für den CHAR-Befehl die Zeichensatzadresse nicht über das TED-Register festgelegt, da im Hires-Modus die entsprechenden Zeichen-Bitkonfigurationen direkt aus dem Zeichensatzspeicher in den Grafikbildschirm übertragen werden. Da es sich hier also nicht um einen Zugriff des TED handelt, spielt es auch keine Rolle, ob das RAM/ROM Bank-Bit gesetzt ist oder nicht (d. h. in Bezug auf das Arbeiten mit der Hires-Grafik spielt es natürlich wohl eine Rolle). Die Adresse für den Zeichensatz für den CHAR-Befehl im Hires-Modus läßt sich aber glückicherweise trotzdem ändern, und zwar ist das Hi-Byte der Zeichensatzadresse im RAM-Speicherzelle $ 02E4 (dez. 740) abgelegt (das Lo-Byte der Adresse ist ja auch im Normalmodus immer 0). Über POKE 740, Hi.Byte der ZS-Adr. läßt sich also de gewünschte Verschiebung der Zeichensatzadresse vornehmen. Der durch das vorgestellte Beispielprogramm (Listing 3) erzeugte veränderte Zeichensatz kann allerdings nur sehr bedingt im Zusammenhang mit Hires-Grafiken verwendet werden, da dieser aus Platzgründen in der Grundversion des C 16/116 nur in einen Bereich gelegt werden konnte, der auch von der Hires-Grafik genutzt wird. Das hat zur Folge, daß Befehle wie z. B. SCNCLR den gesamten Zeichensatz zerstören. Zu beachten ist auch noch, daß nach dem Abschalten der Hires-Grafik, will man mit einem veränderten Zeichensatz weiterarbeiten, das RAM/ROM Bank-Bit wieder gelöscht werden muß, da dieses - wie schon erwähnt - durch das Abschalten gesetzt wird.

Zeitvorteile durch Abschalten des Bildschirmes

Die Erzeugung des Fernseh- bzw. Monitorbildes kostet den C16 relativ gesehen eine Menge Zeit. Dies ist sofort einsichtig, wenn man sich überlegt, was für Datenmengen bei jedem Bildaufbau anfallen: allein für den genutzten Bildschirmausschnitt muß der C16 dem Fernseher insgesamt 64000 (320*200) Bildpunkte "mitteilen", ob sie gesetzt oder ungesetzt sind - von den anfallenden Farb- und Luminanzinformationen einmal abgesehen. Um ein flimmerfreies Bild zu erhalten, wird der Bildschirm pro Sekunde 50 mal neu aufgebaut. Diese ungeheuer komplexe Tätigkeit schlägt selbst bei einem für menschliches Ermessen unvorstellbar schnellen Mikroprozessor kräftig zu Buche und lastet den C16 beispielsweise bis zu einem Drittel aus. Anderseits gibt es aber langwierige Tätigkeiten, bzw. Programmabschnitte, bei denen das Fernsehbild nicht unbedingt nötig ist (Datas einlesen, Grafiken aufbauen, Daten sortieren, und und und ...). Was liegt da näher, als den C16 in diesen Fällen von der Arbeit des Bildaufbaus zu befreien, also auf das Fernsehbild zu verzichten? Da die Zeit, die der C16 am Bildaufbau spart, zur Programmbearbeitung zur Verfügung steht, wird die Programmausführung um bis zu 35 % beschleunigt. Tatsächlich sieht der TED im C16 die Möglichkeit vor, den Bildaufbau zu unterdrücken. Zuständig dafür ist das Bit 4 des TED-Registers 6 (dez. 65286 / hex. $FF06). Ist es gesetzt, ist der Bildschirm eingeschaltet und der C16 erzeugt ein Fernsehbild. Löscht man es, wird der Bildschirm abgeschaltet (auf Hintergrundfarbe gesetzt) und die Verarbeitungsgeschwindigkeit entsprechend gesteigert. Nach dem Abschalten kann man den Bildschirm durch erneutes Setzen des Bits jederzeit wieder einblenden. Von Basic aus wird der Bildschirm durch die Anweisung

POKE 65286, PEEK (65286) AND 239

abgeschaltet, und durch die Anweisung **POKE 65286, PEEK (65286) OR 16** wieder eingeschaltet. Es ist unbedingt zu beachten, daß in dieses Register nur maskiert zu schreiben ist, da die übrigen Bits andere, sehr wichtige Steuerungsaufgaben übernehmen und nicht ohne weiteres verändert werden dürfen.

Das Abfrageregister für Tastatur und Joystick

Die Abfrage der Tastatur und des Joystick organisiert der TED im C16 über das Register 8 (dez. 65288 / hex. $FF08). Dieses Register gibt den Zustand der einzelnen Leitungen des Ein-/ Ausgabeportes des TED-Chips wieder.

Zum einen ist dieser Port als Ausgang der Tastaturmatrix (8*8) geschaltet, so daß mit Hilfe des Registers 8 die Tastatur ausgelesen werden kann, und zum anderen sind die verschiedenen Joystick-Leitungen an diesen Port gekoppelt. Auf die Weise können die Joysticks über dasselbe Register abgefragt werden.

Während die Abfrage der Tastatur zum Standard gehört und weder von Basic noch von Maschinensprache aus eine Schwierigkeit darstellen sollte, soll sie hier nicht näher behandelt werden.

Interessanter ist da die Abfrage der Joysticks. Hier der entsprechende Auszug aus dem Betriebssystem:

```
. BFC1  20 87 9D JSR $9D87   ;Argument des JOY-Befehls holen
. BFC4  CA       DEX         ;um eins erniedrigen
. BFC5  E0 02    CPX #$02    ;größer als eins ???
. BFC7  B0 F5    BCS $BFBE   ;ja, dann "ILLEGAL QUANTITY ERROR"
. BFC9  BD FB BF LDA $BFFB,X ;Offset für entsprechenden Port holen
. BFCC  AA       TAX         ;nach X
. BFCD  78       SEI         ;Interrupt sperren
. BFCE  8E 08 FF STX $FF08   ;Port aktivieren
. BFD1  AD 08 FF LDA $FF08   ;Port auslesen
. BFD4  8E 08 FF STX $FF08   ;Port erneut aktivieren
. BFD7  CD 08 FF CMP $FF08   ;Daten stabil ???
. BFDA  D0 F2    BNE $BFCE   ;nein, dann neu auslesen
. BFDC  58       CLI         ;sonst Interrupt freigeben
. BFDD  49 FF    EOR #$FF    ;Bits invertieren
. BFDF  A8       TAY         ;nach Y
. BFE0  29 0F    AND #$0F    ;unteren vier Bits isolieren
. BFE2  AA       TAX         ;nach X
. BFE3  BD F0 BF LDA $BFF0,X ;entsprechenden Wert laden
. BFE6  C0 0F    CPY #$0F    ;Feuerknopf zusätzlich gedrückt ???
. BFE8  90 02    BCC $BFEC   ;nein, ===>
. BFEA  09 80    ORA #$80    ;ja, dann Bit 7 setzen
. BFEC  A8       TAY         ;nach Y
. BFED  4C 81 9A JMP $9A81   ;Wert an Funktion übergeben
. BFF0
. BFF0  .BY 00 01 05 00 07 08 06 00 03 02 04
. BFFB
. BFFB  .BY FA FD
. BFFB
```

Assemblerlisting aus dem Betriebssystem

Diese Routine können Sie so auch in eigenen Programmen zur Abfrage der Joystickports verwenden. Laden Sie dazu das X-Register mit der entsprechenden Portnummer und springen Sie bei $BFC9 (für Port 1 ist der Wert 0, für Port 2 der Wert 1 zu laden). Wollen Sie die Ports in einer Interruptschleife abfragen, ist die Routine ebenfalls geeignet. Es muß dann nur das CLI in $BFDC entfallen. Als Ergebnis übergibt die Routine im Y-Register einen der Joystickpositionen entsprechenden Wert, den Sie im Handbuch auf Seite 185 finden. Bei Verwendung der Routine als Unterprogramm in eigenen Programmen muß der JMP-Befehl in $BFED natürlich durch in RTS ersetzt werden.

Die Interruptmöglichkeiten des TED

Zunächst zur Klärung des Begriffes "Interrupt": Der Interrupt ist eine hardwaremäßige Unterbrechung des laufenden Programmes, bei der aktuellen die Programmzeiger auf den Stack (Stapelspeicher) gerettet werden und in eine spezielle Interruptroutine verzweigt wird.

Diese Möglichkeit wird vom Betriebssystem genutzt, indem es durch einen sogenannten Rasterzeilen-Interrupt jede 10 Millisekunden (= 0,01 s) das aktuelle Programm unterbricht und die sogenannte ROM-IRQ-Routine mit der Adresse $ CE0E (dez. 52750) anspringt.

Diese Routine organisiert u. a. die Abfrage der Tastatur, das Stellen der internen Uhr und das Halten eines Tones. Nach Abschluß der Routine werden wieder alle Zeiger vom Stack zurückgeholt, d.h. die Konfiguration, die vor dem Durchlaufen der Interruptroutine im Speicher bestand, wird wieder hergestellt, und es wird mit der Ausführung des aktuellen Programmes fortgefahren.

Die spezielle, durch den TED gegebene Interruptmöglichkeit nennt sich IRQ (Interrupt ReQuest; daher auch der Name ROM-IRQ-Routine). Die Behandlung des IRQ kann leider nur über Maschinensprache erfolgen, so daß sich dieser Artikelabschnitt primär an die Maschinensprache-Programmierer wendet.

Der IRQ läßt sich softwaremäßig durch den Assemblerbefehl SEI (Set Interruptflag -IRQ verhindern) unterdrücken oder auch durch den Befehl CLI (Clear Interrupt-flag - IRQ zulassen) wieder ermöglichen. Die Startadresse der IRQ-Routine ist in den RAM-Speicherzellen $ 0314 (Lo-Byte, dez. 788) und $0315 (Hi-Byte, dez. 789) festgelegt, wo sie sich von und für eigene Zwecke abändern läßt.

Das IMR und das IRR

Die IRQ-Quellen lassen sich über das TED-Register 10 (Adr. $ FF0A, dez. 65290) festlegen. Es handelt sich um das Interrupt Mask Register (IMR), in dem ein gesetztes Bit eine IRQ-Möglichkeit bestimmt. Im einzelnen hat das IMR folgenden Aufbau:

Bit 0: Übertragsbit aus TED-Reg. 11
Bit 1: 1 = Rasterzeilen-IRQ, ausgelöst durch TED-Reg. 11
Bit 2: 1 = Impuls vom Lightpen - diese Möglichkeit existiert beim C 16/116 noch nicht
Bit 3: 1 = Unterlauf von Timer 1 (TED- Reg. 01)
Bit 4: 1 = Unterlauf von Timer 2 (TED- Reg. 2/3)
Bit 5: unbenutzt
Bit 6: 1= Unterlauf von Timer 3 (TED- Reg. 4/5)
Bit 7: unbenutzt

Nun gibt es noch ein zweites Register, nämlich das Interrupt Request Register (IRR) mit der Adresse $ FF09 (dez. 65289, TED-Reg. 9), in dem mit dem Eintreten der im IMR bezeichneten Ereignisse ein bestimmtes Bit gesetzt wird. Die Bits 1 bis 6 stimmen also mit denen des IMR überein, nur das ein gesetztes Bit hier nicht einen IRQ-Auslöser festsetzt, sondern kennzeichnet, daß ein bestimmtes Ereignis eingetroffen ist. Bit 0 dieses Registers ist ungenutzt, Bit 7 wird dann gesetzt, wenn ein Bit im IMR gesetzt ist und durch das Eintreten des betreffenden Ereignisses das korrespondierende Bit im IRR ebenfalls gesetzt wird. Genau dann ist auch der Interruptfall gegeben, d.h. es wird ein IRQ ausgelöst.

Die Timer des TED

Der TED verfügt über drei unabhängige 16-Bit-Timer, die alle in der Lage sind, einen IRQ auszulösen.

Diese Timer werden mit jedem Systemtakt (d. h. also etwa jede Mikrosekunde (= 101(-6)s)) um eine Einheit heruntergezählt.

Timer 1 tut sich dabei noch durch eine Besonder-heit hervor, denn im Gegensatz zu den anderen beiden Timern wird er nach dem Unterlauf automatisch wieder auf den gewählten Startwert zurückgesetzt, was ihn befähigt, Zeitintervalle frei wählbarer Länge zu timen. Die anderen beiden Timer erhalten nach einem Unterlauf

```
100 REM *********************
105 REM *                   *
110 REM * DEMO ZUM TIMER-IRQ *
115 REM *                   *
120 REM * 1986 BY D.D.+W.R.  *
125 REM *                   *
130 REM *********************
135 :
140 :
145 FORA=818TO893:READD:P=P+D:POKEA,D:NEXT
150 IFP<>9432THENPRINT"FEHLER IN DEN DATA-ZEILEN!":END
155 PRINT"EINSCHALTEN MIT SYS 818."
160 PRINT"AUSSCHALTEN MIT SYS 844.":END
165 :
170 :
175 DATA120          : EIN   SEI          " INTERRUPT VERHINDERN
180 DATA173,10,255   :       LDA $FF0A    " BIT 4 (BIT FUER TIMER 2) IM
185 DATA9,16         :       ORA #$10     " INTERRUPT MASK REGISTER
190 DATA141,10,255   :       STA $FF0A    " SETZEN
195 DATA169,128      :       LDA #$80     " STARTWERT FUER
200 DATA141,2,255    :       STA $FF02    " TIMER 2
205 DATA141,3,255    :       STA $FF03    " FESTLEGEN (KANN AUCH WEGFALLEN)
210 DATA169,100      :       LDA #$64     " NEUE IRQ-ADRESSE LO-BYTE
215 DATA162,3        :       LDX #$03     " NEUE IRQ-ADRESSE HI-BYTE
220 DATA208,16       :       BNE $SET     " SETZEN
225 DATA234          :       NOP          "
230 DATA234          :       NOP          "
235 DATA234          :       NOP          "
240 DATA120          : AUS   SEI          " INTERRUPT VERHINDERN
245 DATA173,10,255   :       LDA $FF0A    " BIT 4 (BIT FUER TIMER 2) IM
250 DATA41,239       :       AND #$EF     " INTERRUPT MASK REGISTER
255 DATA141,10,255   :       STA $FF0A    " WIEDER LOESCHEN
260 DATA169,14       :       LDA #$0E     " ALTE IRQ-ADRESSE LO-BYTE
265 DATA162,206      :       LDX #$CE     " ALTE IRQ-ADRESSE HI-BYTE
270 DATA141,20,3     : SET   STA $0314    " IRQ-ADRESSE
275 DATA142,21,3     :       STX $0315    " SETZEN
280 DATA88           :       CLI          " INTERRUPT WIEDER ERMOEGLICHEN
285 DATA96           :       RTS          "
290 DATA234          :       NOP          "
295 DATA234          :       NOP          "
300 DATA234          :       NOP          "
305 DATA173,9,255    : IRQ   LDA $FF09    " INTERRUPT REQUEST REGISTER (IRR)
310 DATA41,16        :       AND #$10     " HANDELT ES SICH UM EINEN
315 DATA208,3        :       BNE $TIRQ    " IRQ AUSGELOEST DURCH TIMER 2 ?
320 DATA76,14,206    :       JMP $CE0E    " NEIN, ZUR ROM-IRQ-ROUTINE
325 DATA141,9,255    : TIRQ  STA $FF09    " IRR LOESCHEN
330 DATA162,39       :       LDX #$27     "
335 DATA169,42       :       LDA #$2A     " ERSTE BILDSCHIRMZEILE
340 DATA157,0,12     : FILL  STA $0C00,X  " MIT
345 DATA202          :       DEX          " STERNEN (*) FUELLEN
350 DATA16,250       :       BPL $FILL    "
355 DATA76,195,252   :       JMP $FCC3    " ZUM ENDE DER ROM-IRQ-ROUTINE

READY.
```

Listing 4

immer den Wert #$ FFFF. Der Startwert der Timer läßt sich über die TED-Register 0 bis 5 (Adressen $ FF00 bis $ FF05, dez. 65280 bis 65285) festlegen, wobei Reg. 0 das Lo-Byte und Reg. 1 das HI-Byte des Timers 1 enthält, Reg. 2 das Lo-Byte des Timers 2 u.s.w..

Wird aus diesen Registern gelesen, so erhält man den aktuellen Wert der Timer. Beim Umgang mit den Timern ist zu beachten, daß diese auch vom Betriebssystem zum Erzeugen einer Zufallszahl und für Eingabe/Ausgabe-Vorgänge gebraucht werden.

Während ersteres für uns keine Rolle spielt, führt letzteres sogar dazu, daß Timer 1 für uns als IRQ-Quelle leider ausfällt. Will man einen der beiden anderen Timer für eigene Zwecke als IRQ-Quelle nutzen, muß man in der Initialisie-rungsroutine neben der Festsetzung der neuen IRQ-Adresse das entsprechende Bit im IMR setzen. Damit die IRQ-Adresse überhaupt geändert werden kann, muß die Initialisierungsroutine mit dem Befehl SEI beginnen. Das hat eine einfache Bewandtnis: Beginnen Sie die Routine nicht mit SEI, so kann es unter Umständen passieren, daß sich der Computer bereits nach dem Ändern des Lo- Bytes der IRQ-Adresse verabschiedet, da vielleicht genau nach dieser Änderung (also dann, wenn das HI-Byte noch nicht geändert ist) ein IRQ ausgelöst wird, der zum Anspringen einer falschen Adresse führen würde. Das Setzen des zuständigen Bits im IMR sollte übrigens über einen ORA-Befehl geschehen, damit der Zustand der anderen Bits erhalten bleibt.

In der neuen IRQ-Routine sollte nun zunächst geprüft werden, ob der IRQ überhaupt von einem Timer und nicht durch eine bestimmte Rasterzeile ausgelöst wurde. Dazu wird getestet, ob das dem als IRQ-Auslöser festgelegten Timer entsprechende Bit im IRR gesetzt ist. Ist das nicht der Fall, so handelt es sich um einen Rasterzeilen-IRQ und es muß in die ROM-IRQ-Routine verzweigt werde. Handelt es sich um einen Timer-IRQ, so muß das IRR noch gelöscht werden, damit nicht sofort nach Beendigung der IRQ-Routine wieder ein IRQ ausgelöst wird, was den Computer schließlich und bedauernswerterweise zum Abstürzen bringen würde. Gelöscht wird das IRR, indem der aus ihm herausgelesene Wert auch wieder in es hineingeschrieben wird. Enden muß die eigene Routine mit dem Sprung ans Ende der ROM-IRQ-Routine, wo die auf den Stack gebrachten Parameter zurückgeholt werden und der IRQ mit dem Assemblerbefehl RTI (Return from Interrupt) abgeschlossen wird. Der entsprechende Sprungbefehl lautet JMP $ FCC3. Das Gesagte soll nun noch durch ein kurzes Lernprogramm (Listing 4), das während des Timer-IRQs die erste Zeile mit einem nicht mehr löschbaren Zeichen vollschreibt, untermalt werden.

Der Rasterzeilen-IRQ

Die zweite Möglichkeit einen IRQ auszulösen, ist die über eine bestimmte Rasterzeile. Der Bildschirm eines Fernsehers oder Monitors besteht nämlich aus einer sogenannten Lochrasterplatine, über die Zeile für Zeile, Punkt für Punkt ein von der Elektronik im hinteren Ende der Bildröhre erzeugter Elektronenstrahl gesteuert wird. Je nach den am UHF-Antenneneingang des Fernsehers anliegenden Informationen, die in unserem Falle vom TED stammen, werden dann von diesem Elektronenstrahl einzelne Punkte auf der fluoreszierenden Schicht des Bildschirmes zum Leuchten angeregt. Unter Rasterzeilen versteht man also die einzelnen Punktzeilen dieser Lochrasterplatine. Normalerweise hat ein Fernseher bzw. Monitor 625 Rasterzeilen, wovon uns aber nur etwa 260 zur Verfügung stehen, da der TED aus Geschwindigkeitsgründen nur Halbbilder erzeugt (wenn Sie sich das Bild einmal genau ansehen, werden Sie merken, daß jede zweite Rasterzeile schwarz bleibt) und noch einige Rasterzeilen am oberen und unteren Bildschirmrand "verschluckt" werden. Etwa jeweils 30 Rasterzeilen werden oben und unten vom Bildrahmen beansprucht, so daß der für den Benutzer verfügbare Raum genau 200 Rasterzeilen beträgt. Darin liegt auch die Erklärung, warum der C16/116 im Hires-Modus eine Y-Auflösung von 200 Punkten hat. Aus uns verschlossenen Gründen (es lassen sich nur Vermutungen anstellen) haben jetzt die Jungens von Commodore den Anfang des frei verfügbaren Bildschirmbereiches mit Rasterzeile 3 festgelegt und das Ende dementsprechend mit Rasterzeile 202. Über die TED-Register 29 und 28 Bit 0 (Adressen $ FF1D und $ FF1C, dez. 65309, 65308) kann man erfahren, welche Rasterzeile gerade erstellt wird.

Dabei wird Bit 0 in Register 28 gesetzt, wenn der Rasterzeilenwert in Register 29 den Wert 255 überschreitet. Die restlichen Register 28 sind unbenutzt. Auch ein bestimmter Rasterpunkt innerhalb einer Zeile läßt sich über das TED-Register 30 (Adr. $ FF1E, dez. 65310) erfahren. Da jedoch der gesamte Bildschirm innerhalb von nur 20 Millisekunden und eine einzelne Rasterzeile damit in etwa nur 0,08 Millisekunden aufgebaut wird, ist auch von Maschinensprache aus eine gezielte Abfrage auf eine Rasterzeile genau nur schwer und auf einen einzelnen Rasterpunkt vollkommen unmöglich.

In den TED-Registern 11, (Adr. $ FF0B, dez. 65291) und 10 Bit 0 (Adr. $ FF0A, dez. 65290) läßt sich jedoch eine Rasterzeile festlegen, bei der, wenn auch das entsprechende Bit im IMR gesetzt ist, ein IRQ ausgelöst wird. Bit 0 des Registers 10 ist dabei wieder das Übertragsbit aus Register 11. Der Rasterzeilen-IRQ wird in mehrfacher Hinsiht vom Betriebssystem genutzt. Durch das eben genannte werden nämlich die Rasterzeilen 161 (#$ A1) und 204 (#$ CC) als IRQ-Auslöser festgelegt.

Wie auch schon zu Beginn dieses Artikelabschnittes kurz angesprochen, wird das aktuelle Programm nun, da der Bildschirm in 20 Millisekunden (= 0,02 s) aufgebaut wird und es sich um zwei Unterbrechungen innerhalb dieser Zeit handelt, etwa jede 1/100 Sekunde in seiner Ausführung unterbrochen. Nur bei jedem zweiten Rasterzeilen-IRQ (nämlich bei Zeile 204), also alle 20 Millisekunden, wird aber die Tastatur abgefragt, die interne Uhr getaktet u.s.w..

Dann nämlich wird in eine Unterroutine verzweigt, deren Startadresse in den RAM-Speicherzellen $ 0312 (Lo-Byte, dez. 786) und $ 0313 (Hi-Byte, dez. 787) festgelegt ist und die diese Aufgaben übernimmt. Die interne Uhr wird dabei innerhalb von fünf Unterbrechungen, die in 5 x 0,02 s = 0,1 s stattfinden, um sechs Takte weitergestellt, wobei ein Takt hier einer 1/60 Sekunde entspricht und damit sechs Takte wieder 6/60 s = 0,1 s ergeben. Der Sound wird ebenfalls in dieser Unterroutine bearbeitet, denn durch den IRQ ist es überhaupt erst möglich, einen Ton für eine längere Zeitdauer zu halten, ohne den Programmablauf zu blockieren.

So kann nämlich in der IRQ-Routine, die ja quasi parallel zum aktuellen Programm abläuft, die durch den Sound-Befehl festgelegte Tondauer heruntergezählt und der Ton schließlich ausgeschaltet werden.

Die Möglichkeit, einen der beiden IRQ-Vektoren (Adr. $ 0314/ $ 0315 bzw. $ 0312/ $ 0313) auf die Startadresse einer eigenen IRQ-Routine zu verbiegen, kann natürlich genutzt werden, um auch noch andere Aufgaben innerhalb des Interrupts parallel zum eigentlich laufenden Programm abzuarbeiten. Nach diesem Prinzip arbeitet auch eine Interrupt-Uhr, die während des IRQs die aktuelle Zeit in die rechte obere Bildschirmecke schreibt. Die im Artikelabschnitt über Softscrolling vorgestellte Softscroll-Routine wird ebenfalls auf diese Weise realisiert, so daß das Scrollen parallel zu anderen Handlungen geschehen kann.

Der Rasterzeilen-IRQ wird aber noch auf eine andere Art vom Betriebssystem genutzt, nämlich um die Mischung von Text und Grafik zu ermöglichen. Darin liegt auch die Erklärung, warum bei zwei Rasterzeilen ein IRQ ausgelöst wird, obwohl nur bei jedem zweiten Mal die erwähnten Aufgaben ausgeführt werden, und warum es sich gerade um die Rasterzeilen 161 und 204 handelt. Wird der IRQ im Grafikmodus mit Textzeilen bei Rasterzeile 161 ausgelöst, so wird zunächst kurz gewartet, bis der TED mit dem Bildaufbau bei Zeile 163 angelangt ist, wo dann vom Grafik- in den Textmodus umgeschaltet wird. Der schon erstellte Grafikteil bleibt davon unbeeinflußt, da zunächst das ganze Bild zuende aufgebaut wird, bevor auch der obere Teil geändert wird. Dazu kommt es aber erst gar nicht, da am unteren Bildrand außerhalb des frei verfügbaren Bereiches nach einem erneuten Rasterzeilen-IRQ bei Zeile 204 wieder in den Grafikmodus zurückgeschaltet wird. Will man den Rasterzeilen-IRQ für eigene Zwecke nutzen, d.h. neue Rasterzeilen

als IRQ-Quellen definieren, ohne die ROM-IRQ-Routine auszuschalten, so stößt man auf erhebliche Probleme. Diese liegen darin begründet, daß die ROM-IRQ-Routine leider keine Rücksicht auf unsere Wünsche nimmt und die von uns festgesetzten Rasterzeilen wieder durch die eigenen Werte überschreibt.

Die einzige Lösung liegt darin, daß man die ROM-IRQ-Routine ins RAM kopiert und dort modifiziert. Die ROM-IRQ-Routine liegt im Bereich von $ CE0E bis $ CE5F und in ihr wird zunächst getestet, ob es sich überhaupt um einen Rasterzeilen-IRQ handelt, d. h. ob Bit 2 im IRR gesetzt ist.

Diese Abfrage muß natürlich erhalten bleiben, doch dahinter kann der Test auf eigene Rasterzeilen erfolgen. Listing 5 erzeugt auf diese Weise einen Hintergrund und Rahmen, der zur einen Hälfte schwarz und zur anderen weiß ist. Dazu wird in der Bildschirmmitte bei Rasterzeile 102 ein IRQ ausgelöst, wo dann die Hintergrund- und Rahmenfarbe schwarz gesetzt und, da die Rasterzeilen der ROM-IRQ-Routine mitberückichtigt werden müssen, der nächste Rasterzeilen-IRQ für Zeile 161 festgelegt wird.

Bei Rasterzeile 161 wird also in die kopierte ROM-IRQ-Routine verzweigt. Die nächste Rasterzeile, bei der ein IRQ ausgelöst werden soll, ist, wie auch in der nicht modifizierten ROM-IRQ-Routine, Zeile 204. Dieser Wert kann also erhalten bleiben. Normalerweise würde jetzt wieder Zeile 161 folgen, wenn wir diesen Wert nicht in der kopierten Routine ändern würden; und zwar setzen wir dafür Rasterzeile 240 außerhalb des sichtbaren Bereiches fest. Beim nächsten IRQ gelangen wir also wieder in die eigene Routine, wo nun für die obere Bildschirmhälfte die Hintergrund- und Rahmenfarbe weiß gesetzt wird. Bei Zeile 102 werden dann die Farben für die untere Hälfte wieder gewechselt u.s.w..

Die eigene Routine muß wie auch beim Timer-Interrupt mit einem Sprung ans Ende der ROM-IRQ-Routine (Adr. $ FCC3) enden. Auch darf nicht vergessen werden, vor diesem Sprung das IRR zu löschen. In der Initialisierungsroutine muß übrigens auch der zweite IRQ-Vektor in den RAM-Speicherzellen $ 0312/ $ 0313 (Listing IRQ-Uhr-Vektor genannt)

```
100 REM ****************************
105 REM *                          *
110 REM * ZWEIFARBIGER HINTERGRUND & *
115 REM * RAHMEN                   *
120 REM *                          *
125 REM * 1986 BY D.D.+W.R.         *
130 REM *                          *
135 REM ****************************
140 :
145 :
150 FORA=818TO972:READD:P=P+D:POKEA,D:NEXT
155 IFP<>19201THENPRINT"FEHLER IN DEN DATA-ZEILEN!":END
160 PRINT"IRQ EINSCHALTEN MIT SYS 818."
165 PRINT"AUSSCHALTEN MIT SYS 830.":END
170 :
175 :
180 DATA120          : EIN   SEI          " INTERRUPT VERHINDERN
185 DATA169,3        :       LDA #$03     " NEUE IRQ- & IRQ-UHR-ADRESSE HI-BYTE
190 DATA162,175      :       LDX #$AF     " NEUE IRQ-UHR-ADRESSE LO-BYTE
195 DATA160,86       :       LDY #$56     " NEUE IRQ-AFRESSE LO-BYTE
200 DATA208,10       :       BNE $SET     " NEUE ADRESSEN SETZEN
205 DATA234          :       NOP          "
210 DATA234          :       NOP          "
215 DATA234          :       NOP          "
220 DATA120          : AUS   SEI          " INTERRUPT VERHINDERN
225 DATA169,206      :       LDA #$CE     " ALTE IRQ- & IRQ-UHR-ADRESSE HI-BYTE
230 DATA162,66       :       LDX #$42     " ALTE IRQ-UHR-ADRESSE LO-BYTE
235 DATA160,14       :       LDY #$0E     " ALTE IRQ-ADRESSE LO-BYTE
240 DATA141,19,3     : SET   STA $0313    " IRQ-UHR-ADRESSE UND
245 DATA141,21,3     :       STA $0315    " IRQ-ADRESSE
250 DATA142,18,3     :       STX $0312    " SETZEN
255 DATA140,20,3     :       STY $0314    "
260 DATA88           :       CLI          " INTERRUPT WIEDER ERMOEGLICHEN
265 DATA96           :       RTS          "
270 DATA234          :       NOP          "
275 DATA234          :       NOP          "
280 DATA234          :       NOP          "
285 DATA173,9,255    : IRQ   LDA $FF09    " INTERRUPT REQUEST REGISTER (IRR)
290 DATA41,2         :       AND #$02     " HANDELT ES SICH UM
295 DATA240,40       :       BNE $KRI     " EINEN RASTERZEILEN-IRQ ?
300 :
305 ---------------------- EIGENER TEIL DER IRQ-ROUTINE ------------------------
310 :
315 DATA173,11,255 :         LDA $FF0B    " JA, IRQ-RASTERZEILE IN AKKU
320 DATA162,161    :         LDX #$A1     " NAECHSTE IRQ-RASTERZEILE
325 DATA160,0      :         LDY #$00     " RAHMEN- & HINTERGRUNDFARBE SCHWARZ
330 DATA201,102    :         CMP #$66     " WURDE DER IRQ BEI
335 DATA240,8      :         BEQ $CSET    " RASTERZEILE 102 AUSGELOEST ?
340 DATA162,102    :         LDX #$66     " NEIN, NAECHSTE IRQ-RASTERZEILE
345 DATA160,113    :         LDY #$71     " RAHMEN- & HINTERGRUNDFARBE WEISS
350 DATA201,240    :         CMP #$F0     " WURDE DER IRQ BEI
355 DATA208,18     :         BNE $RIRQ    " RASTERZEILE 240 AUSGELOEST ?
360 DATA142,11,255 : CSET    STX $FF0B    " JA, NAECHSTE IRQ-RASTERZEILE UND
365 DATA140,21,255 :         STY $FF15    " NEUE HINTERGRUND- &
370 DATA140,25,255 :         STY $FF19    " RAHMENFARBE SETZEN
375 DATA173,9,255  :         LDA $FF09    " IRR
380 DATA141,9,255  :         STA $FF09    " LOESCHEN
385 DATA76,195,252 :         JMP $FCC3    " ANS ENDE DER ROM-IRQ-ROUTINE
390 :
395 ------------------------------------------------------------------------
400 :
405 DATA32,96,206  : RIRQ    JSR $CE60    "
410 DATA44,216,7   : KRI     BIT $07D8    "
415 DATA16,14      :         BPL $FLG1    "
420 DATA173,1,253  :         LDA $FD01    "
425 DATA141,212,7  :         STA $07D4    "
430 DATA16,6       :         BPL $FLG1    "
435 DATA32,149,234 :         JSR $EA95    "
440 DATA32,91,234  :         JSR $EA5B    "
445 DATA32,228,227 : FLG1    JSR $E3E4    "
450 DATA173,9,255  :         LDA $FF09    " INTERRUPT REQUEST REGISTER (IRR)
455 DATA41,2       :         AND #$02     " HANDELT ES SICH UM
460 DATA240,40     :         BEQ $FLG2    " EINEN RASTERZEILEN-IRQ ?
465 DATA141,9,255  :         STA $FF09    " JA, IRR LOESCHEN
470 DATA44,11,255  :         BIT $FF0B    " TEST, OB IRQ BEI ZEILE 161 ODER 204
475 DATA169,204    :         LDA #$CC     " NAECHSTE IRQ-RASTERZEILE
480 DATA80,27      :         BVC $FLG3    " --> IRQ BEI RASTERZEILE 161
485 DATA108,18,3   :         JMP ($0312)  " IRQ BEI ZEILE 204, UHR TAKTEN ETC.
490 DATA32,191,207 :         JSR $CFBF    " UHR TAKTEN & REKORDER BEDIENEN
495 DATA32,205,206 :         JSR $CECD    " TON HALTEN ODER GGF. ABSCHALTEN
500 DATA165,251    :         LDA $FB      "
```

Listing 5, 1. Teil

```
505 DATA72          :        PHA          "
510 DATA169,0       :        LDA #$00     "
515 DATA133,251     :        STA $FB      "
520 DATA8           :        PHP          "
525 DATA88          :        CLI          "
530 DATA32,17,219   :        JSR $DB11    " TASTATUR ABFRAGEN (SCNKEY)
535 DATA40          :        PLP          "
540 DATA104         :        PLA          "
545 DATA133,251     :        STA $FB      "
550 DATA169,240     :        LDA #$F0     " NAECHSTE IRQ-RASTERZEILE (GEAENDERT)
555 DATA141,11,255  : FLG3   STA $FF0B    " SETZEN
560 DATA76,190,252  : FLG2   JMP $FCBE    "

READY.
```

Listing 5, 2. Teil

geändert werden, da die Unterroutine, die die Uhr taktet u.s.w., ebenfalls im RAM liegen muß. Das klingt nun alles etwas kompliziert (ist es auch), doch wenn Sie das in den Basic-Loader integrierte Assemblerlisting einmal genau studieren (abzutippen brauchen Sie natürlich nur die DATA-Werte), werden Sie als hochtalentierter Computerfreak sicher schnell zur Erleuchtung gelangen.

Softscrolling

Das Scrollen ist ein Effekt, den Sie - möglicherweise ohne ihn beim Namen zu kennen - mit Sicherheit schon auf Ihrem C16 beobachtet haben. Wenn Sie nämlich beispielsweise ein Programm listen, das länger als eine Bildschirmseite ist, passiert folgendes, sobald das Listing den unteren Bildschirmrand erreicht: Der gesamte Bildschirminhalt wird um eine Zeichenzeile nach oben verschoben, die oberste Bildschirmzeile verschwindet und die unterste, frei werdene Zeile wird die nächste Zeile des Listings geschrieben. Dieses "Rollen" der Zeilen über den Bildschirm nennt man in Computer-Neudeutsch "Scrollen" (vom englischen "Scroll - die Schriftrolle). Beim Scrollen unterscheidet man zum einen das vertikale Scrollen und zum anderen das horizontale Scrollen. Beim Scrollen in vertikaler Richtung bewegt sich der Bildschirminhalt von oben nach unten bzw. von unten nach oben (beispielsweise beim Listen), beim Scrollen in horizontaler Richtung von links nach rechts bzw. von rechts nach links. Durch Kombination beider Bewegungsrichtungen lassen sich Mischformen erzielen, so daß der Eindruck entsteht, der Bildschirminhalt bewege sich diagonal. Ein weiteres Unterscheidungsmerkmal ist die Auflösung des Scrolleffektes; während der Bildschirminhalt beim Listen gleich um eine ganze Zeichenzeile (eine Zeichenzeile besteht aus insgesamt acht Punkt- oder Bildschirmzeilen) gescrollt wird. Wenn der untere Bildschirmrand erreicht ist, wird bei professionellen Spielen beispielsweise meist das sogenannte Softscrolling verwendet, bei dem der Bildschirminhalt punktweise verschoben werden kann. Durch diese achtfach höhere Scrolling-Auflösung gegenüber dem LIST-Effekt wirken die entstehenden Bewegungen viel gleichmäßiger, fließender und weicher - eben soft, womit auch schon der Ursprung des Namens "Softscrolling" eindeutig geklärt wäre. Kaum ein Profispiele-Programmierer läßt diesen äußerst beeindruckenden Grafikeffekt ungenutzt, den der C16 in gleichem Maße wie sein "großer Bruder" C64 beherrscht. Ein Meister auf diesem Gebiet - was den C16 betrifft - ist sicherlich Udo Gertz von Kingsoft, der u. a. Spiele wie TOM oder Winterolympiade programmiert hat, die beide kräftig vom Softscrolling-Effekt Gebrauch machen. Aber auch dem engagierten Hobbyprogrammierer stehen die Softscrolling-Möglichkeiten des C16 für eigene Anwendungen ohne weiteres offen: Man muß halt nur wissen, wie's funktioniert. Ihnen dies zu vermitteln, ist Aufgabe dieses Kapitels. Ein Wort vorweg: sinnvolle Softscrolling-Programmierung ist aus Geschwindigkeitsgründen nur von Maschinensprache aus möglich. Basic-Programmierer seien hier entweder ausschließlich auf das Demoprogramm am Ende des Kapitels oder aber auf weiterführende Literatur zum Thema Assemblerprogrammierung verwiesen.

Für die Steuerung des Scrolling-Effektes sind im TED-Chip des C16 zwei Register zuständig: für das Scrollen in vertikaler Richtung ist Register 6 (dez. 65286 / hex. $FF06), für das Scrollen in horizontaler Richtung das Register 7 (dez. 65287/ hex. $FF07) verantwortlich. Streng genommen sind es jeweils nur die vier unteren Bits beider Register; die übrigen Bits haben andere, sehr wichtige Steuerungsaufgaben. Entsprechend darf in beide Register nur maskiert geschrieben werden, um den TED nicht bei seiner Arbeit zu behindern. Die unteren drei Bits des Registers 6 sind für die Scrollposition des Bildschirmes in Y-Richtung zuständig. Der Normalwert dieser drei Bits beträgt 3, d. h., es sind die Bits 0 und 1 gesetzt, während Bit 2 ungesetzt ist.
Zusammen mit den übrigen Bits des Registers ergibt sich ein Registerinhalt von 27 (hex. #$1D). Sie können dies ganz einfach per PEEK-Befehl überprüfen. Wenn Sie nun mit POKE 65286,28 die Scrollposition des Bildschirmes um eins erhöhen, sehen Sie, was passiert: der gesamte Bildschirminhalt wird um eine Punktzeile nach unten verschoben. Gleichzeitig sollte Ihnen aber auffallen, daß in der obersten Punktzeile ausgesprochener Murks erscheint. Dies ist natürlich weder beabsichtigt noch sehr sinnvoll.
Doch auch hier hat der TED gleich intern eine Lösung parat: das Bit 3 des Registers 6 dient nämlich dazu, den Bildschirm zwischen einem 25- und einem 24-Zeilenmodus umzuschalten.
Normalerweise (nach dem Einschalten) befindet sich der C16 im 25-Zeilenmodus, auf dem Bildschirm werden 25 Zeichenzeilen dargestellt, Bit 3 des Registers 6 ist gesetzt. Löscht man dieses Bit, so wird der Bildschirm oben und unten jeweils um vier Punktzeilen verkleinert. Zweimal vier Punktzeilen ergeben zusammen acht Punktzeilen, also eine Zeichenzeile. Insgesamt ist der Bildschirm somit um eine Zeichenzeile kleiner geworden, es werden nur noch 24 Zeilen dargestellt. Der Nachteil dieses Vorgehens, es paßt ganz einfach weniger auf den Bildschirm, wird durch den Vorteil, bei Verwendung des Scrolling-Effekts ist oben erwähnter Murks nicht mehr sichtbar, voll aufgewogen.

So, nun haben wir eigentlich alle Angaben, die für das Scrollen nötig sind, zusammen; greifen wir uns das Beispiel Scrollen nach unten heraus, um die Programmierung in der Praxis zu verdeutlichen. Nach unseren Erfahrungen empfiehlt sich folgende Vorgehensweise:

1 - Füllen Sie den Bildschirm mit dem gewünschten Inhalt, der gescrollt werden soll.

2 - Verkleinern Sie den Bildschirm auf 24 Zeichenzeilen, indem Sie Bit 3 des Registers 6 löschen. Nicht vergessen: in dieses Register ist unbedingt maskiert zu schreiben, da die übrigen Bits teilweise sehr wichtige Steuerungsaufgaben besitzen und nicht ohne weiteres beeinflußt werden dürfen. Ansonsten kann es sehr leicht passieren, daß Sie den TED und damit den Computer lahmlegen.

3 - Setzen Sie die unteren drei Bits des Registers 6 (Scrollposition des Bildschirmes in Y-Richtung) alle auf 0.

4 - Erhöhen Sie den Inhalt dieser drei Bits schrittweise von 0 auf 7. Dies hat am besten in einer Schleife zu erfolgen, die siebenmal durchlaufen wird. Unter Umständen kann eine kleine Verzögerungsschleife nötig werden - entsprechend der gewünschten Scrollgeschwindigkeit.

5 - An diesem Punkt müßte eigentlich zu Punkt 3 zurückgekehrt werden, um dort fortzusetzen. Würde dies jedoch so ohne weiteres erfolgen, würde der Bildschirminhalt ruckartig in die Normalposition zurückversetzt werden, um dann wiederum um sieben

Scrollpositionen nach unten verschoben zu werden. Dieses "Zittern" des Bildschirmes ist natürlich nicht der Effekt, den wir erzielen wollten; deshalb muß hier noch einmal die Programmierkunst einsetzen: bevor zu Punkt 3 zurückverzweigt wird, wird der gesamte Bildschirminhalt um eine Zeichenzeile (nicht um eine Punktzeile !) nach unten verschoben. Dazu wird Zeile 23 in Zeile 24 kopiert, Zeile 22 in Zeile 23 und so weiter. Dabei kann - je nach gewünschtem Effekt - in die oberste, frei werdende Zeile (Zeile 0) entweder die vormals letzte Zeile (Zeile 24) kopiert werden (so rollt derselbe Bildschirminhalt kontinuierlich), oder aber eine neue Zeile, die sich entsprechend anschließt; dies ergibt den Effekt, als sei der Bildschirminhalt ein Ausschnitt eines viel größeren Bildschirmes.

6 - Kehren Sie an dieser Stelle zu Punkt 3 zurück und setzen Sie dort wieder ein.

Das Ergebnis eines Programmes, das diese Schritte analog auf den Computer umsetzen würde, wäre eine fließende, weiche Scrollbewegung des Bildschirminhalts nach unten. Die Vorgehensweise für das Scrollen nach oben ist im Prinzip gleich, da dasselbe Register bedient werden muß:

1 - Bildschirm füllen.
2 - Bildschirm auf 24 Zeichenzeilen verkleinern.
3 - Setzen Sie die unteren drei Bits des Registers 6 alle auf 1. Dies ergibt einen Wert von sieben.
4 - Erniedrigen Sie den Inhalt dieser drei Bits schrittweise von 7 auf 0.
5 - Verschieben Sie den Bildschirminhalt um eine Zeichenzeile nach oben (umkopieren). Denken Sie daran, daß hier die unterste, frei werdende Zeile einen neuen Inhalt erhalten muß (vgl. scrollen nach unten).
6 - Setzen Sie bei Punkt 3 fort.

Ein entsprechendes Programm ergibt eine fließende Bewegung des Bildschirminhalts nach oben. Dies waren in allen Einzelheiten die beiden möglichen Bewegungsrichtungen in vertikaler Richtung; wie schon oben erwähnt, ist das Ganze auch in horizontaler

```
1000 REM ******************************
1010 REM *                            *
1020 REM * S O F T S C R O L L I N G - *
1030 REM *                            *
1040 REM *            D E M O         *
1050 REM *                            *
1060 REM *  FUER DEN C16/116 & PLUS/4  *
1070 REM *                            *
1080 REM *                            *
1090 REM *                            *
1100 REM *  (W)'86 BY  D.D.  &  W.R.   *
1110 REM *                            *
1120 REM ******************************
1130 :
1140 :
1150 : REM    *** DATAS FUER MASCHINENROUTINE ***
1160 :
1170 DATA A2,27     : REM        LDX #$27      ; TEXTLAENGE DER KOPFZEILE
1180 DATA BD,90,3F  : REM FLG01  LDA $3F90,X   ; KOPFZEILE IN DAS VIDEORAM
1190 DATA 9D,00,0C  : REM        STA $0C00,X   ; KOPIEREN
1200 DATA CA        : REM        DEX           ;
1210 DATA 10,F7     : REM        BPL $FLG01    ;
1220 DATA E8        : REM        INX           ; X=0 -> LINKE, OBERE WINDOW-ECKE
1230 DATA 8E,E7,07  : REM        STX $07E7     ; AUF (0,1) SETZEN
1240 DATA E8        : REM        INX           ; X=1
1250 DATA 8E,E6,07  : REM        STX $07E6     ;
1260 DATA 86,D8     : REM        STX $D8       ; MOMENTANE SCROLLPOSITION AUF 0
1270 DATA A2,22     : REM        LDX #$22      ; NEUER INTERRUPT-VEKTOR LOW
1280 DATA A0,3F     : REM        LDY #$3F      ;                          HIGH
1290 DATA 78        : REM        SEI           ; INTERRUPT SPERREN
1300 DATA 8E,14,03  : REM        STX $0314     ; NEUEN INTER.-VEKTOR SETZEN LOW
1310 DATA 8C,15,03  : REM        STY $0315     ;                            HIGH
1320 DATA 58        : REM        CLI           ; INTERRUPT FREIGEBEN
1330 DATA 60        : REM        RTS           ; ZURUECK INS BASIC
1340 DATA AD,09,FF  : REM        LDA $FF09     ; INTERRUPT-FLAG-REGISTER LESEN
1350 DATA 8D,09,FF  : REM        STA $FF09     ;                     UND LOESCHEN
1360 DATA 29,02     : REM        AND #$02      ; RASTER-INTERRUPT ???
1370 DATA D0,03     : REM        BNE $FLG02    ; ==> JA
1380 DATA 4C,18,CE  : REM        JMP $CE18     ; ==> NORMALER SYSTEM-INTERRUPT
1390 DATA AE,1D,FF  : REM FLG02  LDX $FF1D     ; RASTERZEILE LESEN
1400 DATA E0,50     : REM        CPX #$50      ; INTERRUPT IN ZEILE $FA ???
1410 DATA B0,0E     : REM        BCS $FLG03    ; ==> JA
1420 DATA AD,07,FF  : REM        LDA $FF07     ; STANDARD-BILDSCHIRM EINSCHALTEN
1430 DATA 29,F0     : REM        AND #$F0      ; (40-SPALTEN MODUS UND NORMALE)
1440 DATA 09,08     : REM        ORA #$08      ; (SCROLLPOSITION              )
1450 DATA 8D,07,FF  : REM        STA $FF07     ;
1460 DATA A2,FA     : REM        LDX #$FA      ; NAECHSTER INTERRUPT IN ZEILE $FA
1470 DATA D0,3F     : REM        BNE $FLG04    ; ==> IMMER
1480 DATA C6,D8     : REM FLG03  DEC $D8       ; SCROLLPOSITION ERNIEDRIGEN
1490 DATA 10,19     : REM        BPL $FLG05    ; ==> NOCH GROESSER NULL
1500 DATA AD,00,0C  : REM        LDA $0C00     ; KOPFZEILE UM EIN ZEICHEN NACH
1510 DATA 48        : REM        PHA           ; LINKS VERSCHIEBEN
1520 DATA A2,00     : REM        LDX #$00      ;
1530 DATA BD,01,0C  : REM FLG06  LDA $0C01,X   ;
1540 DATA 9D,00,0C  : REM        STA $0C00,X   ;
1550 DATA E8        : REM        INX           ;
1560 DATA E0,27     : REM        CPX #$27      ;
1570 DATA 90,F5     : REM        BCC $FLG06    ;
1580 DATA 68        : REM        PLA           ;
1590 DATA 8D,27,0C  : REM        STA $0C27     ;
1600 DATA A9,07     : REM        LDA #$07      ; SCROLLPOSITION AUF SIEBEN
1610 DATA 85,D8     : REM        STA $D8       ;
1620 DATA AD,07,FF  : REM FLG05  LDA $FF07     ; BILDSCHIRM AUF 38 SPALTEN
1630 DATA 29,F0     : REM        AND #$F0      ; VERKLEINERN UND SCROLLPOSITION
1640 DATA 05,D8     : REM        ORA $D8       ; INS REGISTER UEBERTRAGEN
1650 DATA 8D,07,FF  : REM        STA $FF07     ;
1660 DATA 20,BF,CF  : REM        JSR $CFBF     ; ==> DATASSETTE, UHR BEDIENEN
1670 DATA 20,CD,CE  : REM        JSR $CECD     ; ==> SOUND
1680 DATA A5,FB     : REM        LDA $FB       ; MODULKONFIGURATION
1690 DATA 48        : REM        PHA           ; MERKEN
1700 DATA A9,00     : REM        LDA #$00      ; ROM
1710 DATA 85,FB     : REM        STA $FB       ; EINSCHALTEN
1720 DATA 08        : REM        PHP           ;
1730 DATA 58        : REM        CLI           ; INTERRUPT SPERREN
1740 DATA 20,11,DB  : REM        JSR $DB11     ; TASTATUR ABFRAGEN
1750 DATA 28        : REM        PLP           ;
1760 DATA 68        : REM        PLA           ; MODULKONFIGURATION
1770 DATA 85,FB     : REM        STA $FB       ; WIEDERHERSTELLEN
1780 DATA A2,0A     : REM        LDX #$0A      ; NAECHSTER INTERRUPT IN ZEILE $0A
1790 DATA 8E,0B,FF  : REM FLG04  STX $FF0B     ;
1800 DATA 4C,BE,FC  : REM        JMP $FCBE     ; ==> RTI
```

Listing 6, Teil 1

```
1810 DATA EA        : REM        NOP          ;
1820 DATA EA        : REM        NOP          ;
1830 DATA EA        : REM        NOP          ;
1840 DATA EA        : REM        NOP          ;
1850 DATA EA        : REM        NOP          ;
1860 DATA EA        : REM        NOP          ;
1870 DATA EA        : REM        NOP          ;
1880 DATA A0,A0,A0,A0,A0,AA,A0,89 : REM TEXT   "     * I"
1890 DATA 8E,94,85,92,92,95,90,94 : REM       "NTERRUPT"
1900 DATA AD,A0,A6,A0,93,8F,86,94 : REM       "- & SOFT"
1910 DATA 93,83,92,8F,8C,8C,89,8E : REM       "SCROLLIN"
1920 DATA 87,AD,84,85,8D,8F,A0,AA : REM       "G-DEMO *"
1930 :
1940 :
1950 : REM    *** EINLESEN DER MASCHINEN-ROUTINE ***
1960 :
1970 POKE52,61:POKE56,61:CLR
1980 :
1990 RESTORE:FORI=16128TO16311:READA$:A=DEC(A$):P=P+A:POKEI,A:NEXTI
2000 IFP<>23828THENPRINT"▒▒▒FEHLER IN DEN DATA-ZEILEN!":END
2010 PRINT"▒▒▒BITTE SEHR ...":SYS16128
READY.
```

Listing 6, Teil 2

Richtung möglich. Dafür ist Register 7 (dez. 65287 / hex. $FF07) des TED zuständig: Analog zu Register 6 geben die drei unteren Bits die Scrollposition des Bildschirminhalts und Bit 3 die Bildschirmgröße an. Ist Bit 3 gesetzt, werden 40 Bildschirmspalten dargestellt ("Normalfall" - nach dem Einschalten), setzt man das Bit auf 0, so werden nur noch 38 Spalten dargestellt. Wie oben ist dieses Verkleinern des Bildschirmes beim Scrollen unverzichtbar. Die Darstellung der Scrollposition entspricht genau der Darstellung in Register 6: Ein Erhöhen der Scrollposition verschiebt den Bildschirminhalt punktweise nach rechts, ein Erniedrigen der Scrollposition entsprechend nach links. Folglich empfiehlt sich beim Scrollen in horizontaler Richtung folgende Vorgehensweise (die Werte im Text beziehen sich auf das Scrollen nach rechts, die in Klammern auf das Scrollen nach links):

1 - Füllen Sie den Bildschirm mit dem gewünschten, zu scrollenden Inhalt.

2 - Verkleinern Sie den Bildschirm durch Löschen des Bits 3 des Registers 7 auf 38 Spalten.

3 - Setzen Sie die unteren drei Bits des Registers 7 alle auf 0 (1). Dies ergibt einen Wert von 0 (7).

4 - Erhöhen (Erniedrigen) Sie den Inhalt dieser drei Bits von 0 (7) auf 7 (0).

5 - Verschieben Sie den Bildschirminhalt um eine Zeichenspalte nach rechts (links). Vergessen Sie nicht, die frei werdende Spalte neu aufzufüllen.

6 - Fahren Sie mit Punkt 3 fort.

Auf diese Weise läßt sich auch ein weiches Scrollen in horizontaler Richtung leicht programmieren. Mischt man beide Bewegungsrichtungen miteinander (beispielsweise abwechselnd um einen Punkt nach rechts und dann um einen Punkt nach unten), so lassen sich auch weiche Bewegungen in dialgonaler Richtung auf die gleiche Art bewerkstelligen. So lassen sich zum Beispiel recht realistische Bewegungen über eine Landschaft programmieren. Jeder, der schon einmal versucht hat, Tom im gleichnamigen Kingsoft-Spiel durch die diversen Bilder zu manövrieren, kennt diesen Effekt. In unserem Demoprogramm haben wir uns das Beispiel der horizontalen Scrollbewegung herausgegriffen, um die entsprechenden Programmiertechniken auch einmal in der Praxis zu veranschaulichen. Geben Sie bitte Listung 6 ein (REM-Anweisungen können Sie dabei ersatzlos weglassen), und starten Sie das Programm mit RUN. Falls Sie bei der Eingabe der Datazeilen keinen Fehler gemacht haben, erscheint in der obersten Bildschirmzeile der Text "INTERRUPT - UND SOFTSCROLLING-DEMO", und dieser Text wird nun punktweise nach links über den Bildschirm gescrollt. Dieser Laufschrifteffekt ist auch bei Profiprogrammierern äußerst beliebt und findet sich in sehr vielen Spielprogrammen wieder.

Nun zur Programmiertechnik: damit beim Scrollen nicht der ganze Bildschirm mitbewegt wird, war der Einsatz eines Rasterzeileninterrupts unumgänglich. Dadurch wird der Bildschirm quasi in zwei Teile aufgespalten: auf der einen Seite Zeile 0, auf der anderen die übrigen 24 Zeilen. Diese beiden Teile des Bildschirmes können völlig unabhängig voneinander gesteuert werden. Doch ist dies an dieser Stelle eigentlich nur ein nebensächlicher Eingriff; wer sich besonders für die Interruptprogrammierung interessiert, möge das Assemblerlisting genau studieren - die Erläuterungen sollten ausreichen, um es verstehen und nachvollziehen zu können. Zur Erklärung des Softscrolling sind eigentlich nur die Zeilen 1480 bis 1650 von Bedeutung. In Zeile 1480 wird die momentane Scrollposition, die im Hilfsregister $D8 zwischengespeichert ist, um 1 erniedrigt. Befindet sich der Wert der Scrollposition noch im erlaubten Bereich (0 bis 7), wird direkt nach Zeile 1620 verzweigt, wo der Bildschirm verkleinert (auf 38 Spalten) und die Scrollposition (maskiert!) in das Register 7 übertragen wird. Dadurch bewegt sich die Kopfzeile jeweils um einen Punkt nach links. Wird der erlaubte Bereich unterschritten, also der Grenzwert 0 erreicht, muß die gesamte Zeile um ein Zeichen nach links verschoben werden (1500 bis 1590).

Dazu wird das Zeichen ganz links auf den Stack geschoben (1500 bis 1510), die Zeile umkopiert (1520 bis 1570) und anschließend das Zeichen vom Stack in die Position ganz rechts geschrieben (1580 bis 1590). Die Scrollposition wird anschließend auf 7 hochgesetzt und in das Register 7 übertragen. Auf diese Weise wird der Inhalt der obersten Bildschirmzeile immer wieder um eine Punktbreite nach links verschoben, ohne daß sich der Zeileninhalt ändert. Es entsteht der Effekt einer Laufschrift. Die Bewegung wirkt dabei fließend und weich, was dem ganzen einen professionellen Anstrich verleiht. Durch die Definition eines entsprechenden Windows ist die scrollende Bildschirmzeile vor dem Überschreiben geschützt.

Umschalten zwischen PAL- und NTSC-Norm

PAL (Phase Alternation Line) und NTSC (National Television System Committee) bezeichnen zwei verschiedene Systeme zur Erzeugung eines Fernsehbildes. "Zwei verschiedene Systeme" ist dabei sogar schon stark übertrieben, denn erzeugt wird das Fernsehbild nach dem gleichen Prinzip, nur sind bei beiden Systemen die Parametereinstellungen (Frequenz, Anzahl der Rasterzeilen) etwas unterschiedlich. Unser - wie wohl auch Ihr - C16 ist auf die in Deutschland gebräuchliche PAL-Norm voreingestellt, so daß man ihn ohne weiteres an ein handelsübliches Fernsehgerät anschließen kann. In Amerika und Japan ist dies anders: Dort wird mit der NTSC-Norm gearbeitet, und entsprechend sind die Computer dort wohl auch eingestellt. Unser C16 bzw. der TED-Chip im C16, der unter anderem für die Erzeugung des Fernsehbildes zuständig ist, kann alle beide Systeme bedienen. Als Schaltbit dient Bit 6 des Registers 7 (dez. 65287 / hex. $FF07). Ist es gesetzt, wird ein NTSC-Bild erzeugt, ist es ungesetzt ein PAL-Bild. Durch Auslesen des Registerinhaltes können Sie sich davon überzeugen, daß das Bit bei Ihrem C16 - entsprechend der in Deutschland üblichen PAL-Norm - nicht gesetzt ist. Wenn Sie das Bit probehalber einmal auf 1 setzen (mit POKE 65287, PEEK (65287) OR 64), sehen Sie, daß das Bild völlig vermurkst wird. Der Fernseher versteht die Signale aus dem Computer einfach nicht mehr.

Uns ist zu diesem Bit keine halbwegs sinnvolle Anwendung eingefallen, was die Wichtigkeit, es zu kennen, keineswegs schmälert. Vielleicht kann ja einer der Leser mit einer Anwendung aufwarten - obwohl sich das wahrscheinlich schwierig gestaltet.

```
1000 REM ******************************
1010 REM *                            *
1020 REM * H A R D W A R E C U R S O R *
1030 REM *                            *
1040 REM *            D E M O         *
1050 REM *                            *
1060 REM *  FUER DEN C16/116 & PLUS/4  *
1070 REM *                            *
1080 REM *                            *
1090 REM *                            *
1100 REM *  (W)'86 BY  D.D.  &  W.R.   *
1110 REM *                            *
1120 REM ******************************
1130 :
1140 :
1150 : REM     *** DATAS FUER MASCHINENROUTINE ***
1160 :
1170 DATA A2,0D     : REM        LDX #$0D       ; NEUE INTERRUPTADRESSE LOW -BYTE
1180 DATA A0,04     : REM        LDY #$04       ;                       HIGH-BYTE
1190 DATA 78        : REM        SEI            ; INTERRUPT SPERREN
1200 DATA 8E,14,03  : REM        STX $0314      ; NEUE I.-ADRESSE SPEICHERN LOW
1210 DATA 8C,15,03  : REM        STY $0315      ;                            HIGH
1220 DATA 58        : REM        CLI            ; INTERRUPT FREIGEBEN
1230 DATA 60        : REM        RTS            ; ZURUECK INS BASIC
1240 DATA AD,0C,FF  : REM        LDA $FF0C      ; HIGH-BYTE DES HARDWARECURSORS
1250 DATA 29,03     : REM        AND #$03       ; UNTEREN BEIDEN BITS ISOLIEREN
1260 DATA AE,0D,FF  : REM        LDX $FF0D      ; AUF BILDSCHIRM SCHREIBEN
1270 DATA 8D,26,0C  : REM        STA $0C26      ; LOW-BYTE DER HARDWARECURSORS
1280 DATA 8E,27,0C  : REM        STX $0C27      ; AUF BILDSCHIRM SCHREIBEN
1290 DATA 4C,0E,CE  : REM        JMP $CE0E      ; ==> NORMALE INTERRUPT-ROUTINE
1300 :
1310 :
1320 : REM     *** EINLESEN DER MASCHINEN-ROUTINE ***
1330 :
1340 RESTORE:FORI=1024TO1053:READA$:A=DEC(A$):P=P+A:POKEI,A:NEXTI
1350 IFP<>2578THENPRINT"\FEHLER IN DEN DATA-ZEILEN!":END
1360 PRINT"\BITTE SEHR ...":SYS1024 READY.
```

Listing 7

Der Hardware - Cursor

Anders als beispielsweise beim C64 wird beim C16 das Blinken des Eingabecursors nicht soft-, sondern vielmehr hardwaremäßig erzeugt. Aus diesem Grunde muß sich der TED die momentane Cursorposition laufend mitprotokollieren. Dies geschieht in den beiden TED-Registern 12 (dez. 65292 / hex. $FF0C) und 13 (dez. 65293 / hex. $FF0D). Dabei wird die Cursorposition nicht im gewohnten X/Y-Koordinatenformat (Spalte/Zeile) dargestellt, sondern es wird die Adresse des Cursors relativ zur Basisadresse des Videoram angegeben. Das Videoram ist ja bekanntlich so organisiert, daß man von links oben bis rechts unten die Bildschirmpositionen zeilenweise durchzählt. Da der Bildschirm 40*25, also 1000 Zeichen faßt, braucht man 10000 Speicherzellen, um alle Bildschirmpositionen darstellen zu können. Um 1000 Positionen adressieren zu können, sind in der Binärdarstellung zehn Bits notwendig. Diese hat der TED in den besagten Registern reserviert: Regiter 13 enthält die acht unteren Bits und die Bits 0 und 1 des Registers 12 stellen die beiden höchstwertigsten Bits der Adresse dar. Die übrigen Bits des Registers 12 haben keine weitere Bedeutung und sind grundsätzlich immer auf 1 gesetzt. Unabhängig von der Lage des Videorams (normal: dez. 3072 / hex. $0C00) wird in diesen Registern die Cursorposition innerhalb des Videorams dargestellt, wobei quasi eine Basisadresse von 0 zugrunde gelegt wird. Diese beiden Register sind natürlich nur in Betrieb, wenn der Cursor wirklich blinkt, man sich also im Direktmodus befindet. Sonst sind alle Bits in den Registern auf 1 gesetzt. Entsprechend schwierig gestaltet sich natürlich das Auslesen der Register; denn sobald man beispielsweise versucht, per PEEK-Befehl auszulesen, erhält man den nichtssagenden Wert 255 (alle Bits gesetzt) als Registerinhalt, da bei Eingabe des Befehls der Computer den Direktmodus verläßt, um den Befehl auszuführen. Hier bietet sich die Abfrage per Interrupt an. Geben Sie dazu bitte Listing 7 ein und starten es mit RUN. Es wird ein kurzes Maschinenprogramm eingelesen und gestartet. Darauf werden in der rechten oberen Bildschirmecke zwei Zeichen dargestellt. Diese kommen dadurch zustande, daß per Interrupt laufend die Inhalte der Register 12 und 13 ausgelesen und in die besagten Bildschirmpositionen geschrieben werden. Anhand dieser Zeichen kann man nun überprüfen, wie Cursorposition und Registerinhalte genau zusammenhängen. Fahren Sie mit dem Cursor einmal willkürlich über den Bildschirm; Sie werden sehen, wie sich die Zeichen verändern. Über die Bildschirmcodes der angezeigten Zeichen kann man die Cursorposition leicht errechnen.

Die Soundmöglichkeiten des TED

Der TED-Chip im C16/116 und Plus/4 besitzt zwei Tongeneratoren, von denen einer nur Töne, der andere sowohl Töne als auch Geräusche erzeugen kann. Beide Tongeneratoren werden in ihren Soundfähigkeiten durch das Basic 3.5 ausreichend unterstützt (SOUND, VOL). Will man die Tongeneratoren jedoch von Maschinensprache aus bedienen oder bestimmte Soundeffekte erzeugen, so ist es unerläßlich, die entsprechenden TED-Register und ihre Bedeutung genau zu kennen. Insgesamt fünf Register sind an der Sounderzeugung des TED beteiligt. Es sind dies die Register 14 bis 18 (dez. 65294 bis 65298, hex. $FF0E bis $FF12). Anders als beispielsweise beim SID des C64 ist die Tonerzeugung beim TED extrem einfach gehalten. Für einen Ton sind nur zwei elementare Parameter, nämlich die Tonhöhe und die Lautstärke, notwendig.

Die Angabe der Lautstärke gilt dabei gleichzeitig für beide Tongeneratoren, so daß sie nicht unabhängig voneinander geregelt werden können. Zuständig sind für die Lautstärke die Bits 0 bis 3 des Registers 17 (dez. 65297 / hex. $FF11). Da auch die übrigen Bits dieses Registers Steuerungsaufgaben besitzen, ist in dieses Register unbedingt maskiert zu schreiben, damit die Bits 4 und 7 nicht beeinflußt werden. Mit den Bits 0 bis 3 ließe sich die Lautstärke rein theoretisch in sechszehn Stufen unterteilen (vier Bits können sechszehn Zustände darstellen). Benutzt werden allerdings nur die ersten neun Stufen (also die Werte 0 bis 8), wobei der Wert 0 für absolute Stille, und der Wert 8 für die maximale Lautstärke steht. Werte größer als 8 (also die Werte 9 bis 15) ergeben gegenüber dem Wert 8 keine weitere Lautstärkensteigerung. Dieses Register kann natürlich nur eine relative Abstufung der Lautstärke steuern. Welche Lautstärke den einzelnen Bytewerten entspricht, entscheidet letztendlich die Stellung des Lautstärkereglers an Ihrem Fernseher bzw. Monitor. Überprüfen Sie also zu allererst immer diese Einstellung, falls Ihr Computer einmal völlig stumm bleiben sollte.

Die zweite zur Tonerzeugung auf dem C16 notwendige Größe ist die Tonhöhe bzw. -frequenz. Der TED stellt zur Darstellung der Frequenz jedes Tongenerators jeweils zehn Bits zur Verfügung. Da sich mit zehn Bits exakt 1024 verschiedene Werte darstellen lassen (0 bis 1023), können diese Werte nicht identisch mit den eigentlichen Frequenzwerten sein. Die entsprechende Umrechnungsformel sowie eine Tabelle mit der Zuordnung "Note - Soundregisterwert - Frequenz" finden Sie im Handbuch auf Seite 211.

Der dort angedruckten Tabelle ist eigentlich wenig hinzuzufügen; für Maschinensprachprogrammierer ist es sinnvoll, sich neben die dort abgedruckten Werte die Hex-Entsprechungen zu notieren - dies erspart einem ständiges Umrechnen. Da sich zehn Bits bekanntlich nicht in einem Byte darstellen lassen, spaltet der TED die Frequenzwerte auf: Die unter

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

100 REM ******************************      <211>
110 REM *                            *      <207>
120 REM *   S O U N D - T E S T E R  *      <235>
130 REM *                            *      <227>
140 REM *   FUER DEN C1/16 & PLUS/4  *      <239>
150 REM *                            *      <247>
160 REM *                            *      <1>
170 REM *   (W)'86 BY  D.D. &  W.R.  *      <86>
180 REM *                            *      <21>
190 REM ******************************      <45>
200 :                                       <2>
210 : REM VARIABLENDEFINITION               <121>
220 :                                       <22>
230 SCNCLR:COLOR0,1:COLOR4,1:COLOR1,6,4:GOSUB910   <92>
240 W%(1)=0:W%(2)=0:W%(3)=0:W%(4)=0:W%(5)=0:M=1    <186>
250 POKE65294,0:POKE65295,0:POKE65296,0:POKE6529
7,0:POKE65298,PEEK(65298)AND252             <158>
260 GOSUB500:GOTO410                        <72>
270 :                                       <73>
280 : REM HAUPTSCHLEIFE                     <249>
290 :                                       <93>
300 GETKEYG$:G=ASC(G$)                      <165>
310 D=1:IFM=3ORM=5THEND=10                  <138>
320 IFG= 29THENW%(M)=W%(M)+D:GOTO480        <100>
330 IFG=157THENW%(M)=W%(M)-D:GOTO480        <141>
340 IFG=145THENGOSUB440:M=M-1:GOTO410       <180>
350 IFG= 17THENGOSUB440:M=M+1:GOTO410       <250>
360 IFG= 88THENEND                          <250>
370 GOTO300                                 <178>
380 :                                       <183>
390 : REM EINSTELLGROESSE AENDERN           <171>
400 :                                       <203>
410 IFM<1THENM=5:ELSEIFM>5THENM=1           <229>
420 FORT=3354+M*80TO3354+M*80+11:POKET,PEEK(T)OR
128:NEXTT                                   <46>
430 GOTO300                                 <238>
440 FORT=3354+M*80TO3354+M*80+11:POKET,PEEK(T)AN
D127:NEXTT:RETURN                           <145>
450 :                                       <253>
460 : REM PARAMETER AENDERN                 <179>
470 :                                       <17>
480 ONMGOSUB540,600,680,750,840             <244>
490 GOSUB500:GOTO300                        <31>
500 FORI=0TO4:CHAR1,3+7*I,23,"{SPACE}":PRINTUSIN
G"###";PEEK(65294+I):NEXT                   <63>
510 :                                       <57>
520 : REM LAUTSTAERKE                       <86>
530 :                                       <78>
540 IFW%(1)<0THENW%(1)=8:ELSEIFW%(1)>8THENW%(1)=
0                                           <32>
550 POKE65297,(PEEK(65297)AND240)ORW%(1)    <200>
560 CHAR1,15,9,"{SPACE}":PRINTUSING"####";W%(1):
RETURN                                      <16>
570 :                                       <118>
580 : REM TONGENERATOR #1                   <81>
590 :                                       <138>
600 IFW%(2)<0THENW%(2)=1:ELSEIFW%(2)>1THENW%(2)=
0                                           <126>
610 CHAR1,15,11,"{SPACE}"                   <182>
620 IFW%(2)=0THENPOKE65297,PEEK(65297)AND239:PRI
NT"{SPACE}AUS"                              <11>
630 IFW%(2)=1THENPOKE65297,PEEK(65297)OR16:PRINT
"{SPACE2}AN"                                <179>
640 RETURN                                  <16>
650 :                                       <198>
660 : REM FREQUENZ STIMME #1                <8>
670 :                                       <218>
680 IFW%(3)<0THENW%(3)=1023:ELSEIFW%(3)>1023THEN
W%(3)=0                                     <75>
690 H%=INT(W%(3)/256):L%=W%(3)-H%*256       <207>
700 POKE65294,L%:POKE65298,(PEEK(65298)AND252)OR
H%                                          <148>
710 CHAR1,15,13,"{SPACE}":PRINTUSING"####";W%(3)
:RETURN                                     <105>
720 :                                       <12>
730 : REM TONGENERATOR #2                   <247>
740 :                                       <32>
750 IFW%(4)<0THENW%(4)=2:ELSEIFW%(4)>2THENW%(4)=
0                                           <206>
760 CHAR1,15,15,"{SPACE}"                   <108>
770 IFW%(4)=0THENPOKE65297,PEEK(65297)AND159:PRI
NT"{SPACE}AUS"                              <200>
780 IFW%(4)=1THENPOKE65297,(PEEK(65297)AND159)OR
32:PRINT"{SPACE}TON"                        <160>
790 IFW%(4)=2THENPOKE65297,(PEEK(65297)AND159)OR
64:PRINT"{SPACE}RAU"                        <254>
800 RETURN                                  <177>
810 :                                       <103>
820 : REM FREQUENZ STIMME #2                <187>
830 :                                       <123>
840 IFW%(5)<0THENW%(5)=1023:ELSEIFW%(5)>1023THEN
W%(5)=0                                     <142>
850 H%=INT(W%(5)/256):L%=W%(5)-H%*256       <176>
860 POKE65295,L%:POKE65296,H%               <45>
870 CHAR1,15,17,"{SPACE}":PRINTUSING"####";W%(5)
:RETURN                                     <94>
880 :                                       <173>
890 : REM BILDAUFBAU                        <19>
900 :                                       <193>
910 A$="*************************************"   <32>
920 B$="*{SPACE37}*"                        <123>
930 FORI=0TO24:CHAR1,0,I,B$:NEXTI           <216>
940 CHAR1,0,0,A$:CHAR1,0,6,A$:CHAR1,0,7,A$:CHAR1
,0,24,A$:CHAR1,0,19,A$                      <0>
950 CHAR1,6,03,"S{SPACE}O{SPACE}U{SPACE}N{SPACE}
D{SPACE2}-{SPACE2}T{SPACE}E{SPACE}S{SPACE}T{SPAC
E}E{SPACE}R"                                <211>
960 CHAR1,2,09,"LAUTSTAERKE{SPACE}:{SPACE4}0{SPA
CE}(0-8)"                                   <25>
970 CHAR1,2,11,"STIMME{SPACE}#1{SPACE3}:{SPACE2}
AUS{SPACE}(AUS/AN)"                         <237>
980 CHAR1,2,13,"FREQUENZ{SPACE}S1{SPACE}:{SPACE4
}0{SPACE}(0-1023)"                          <99>
990 CHAR1,2,15,"STIMME{SPACE}#2{SPACE3}:{SPACE2}
AUS{SPACE}(AUS/TON/RAUSCH)"                 <175>
1000 CHAR1,2,17,"FREQUENZ{SPACE}S2{SPACE}:{SPACE
4}0{SPACE}(0-1023)"                         <167>
1010 CHAR1,3,21,"65294{SPACE2}65295{SPACE2}65296
{SPACE2}65297{SPACE2}65298"                 <160>
1020 CHAR1,3,22,"-----{SPACE2}-----{SPACE2}-----
{SPACE2}-----{SPACE2}-----"                 <129>
1030 RETURN                                 <152>
  ENDE DES LISTINGS
```

Listing 8

en acht Bits (Bit 0 bis 7) werden in einem gesonderten Register abgelegt, und für die beiden oberen Bits (Bit 8 und 9) werden in einem anderen Register zwei Bits reserviert. So enthält Register 14 (dez. 65294 / hex. $FF0E) die unteren acht Bits der Frequenz des ersten Tongenerators, und die Bits 0 und 1 des Registers 18 (dez. 65298 / hex. $FF12) stellen die beiden höchstwertigsten Bits der Frequenz dar. Da die übrigen Bits des Registers 18 andere, sehr wichtige Steuerungsaufgaben übernehmen, ist in dieses Register ausschließlich maskiert zu schreiben. Die Frequenz des zweiten Tongenerators wird entsprechend aufgespalten im Register 15 (dez. 65295 / hex. $FF0F) und in den Bits 0 und 1 des Registers 16 (dez. 65296 / hex. $FF10) abgelegt. Die übrigen Bits des Registers 16 sind bedeutungslos, so daß sie beliebig überschrieben werden können; beim Lesen sind diese Bits immer aus 1 gesetzt. Nachdem die grundlegenden Parameter für die Tonerzeugung festgelegt sind, kommen wir nun zu den eigentlichen Steuerbits. Damit sind die Bits 4 bis 7 des Registers 17 (dez. 65297 / hex. $FF11) gemeint. Mittels dieser Bits lassen sich die beiden Tongeneratoren unabhängig voneinander ein- und abschalten. Bit 4 übernimmt dabei die Steuerung des Tongenerators 1. Bei gesetztem Bit ist der Tongenerator eingeschaltet, erzeugt also einen den Parametern entsprechenden Ton. Dieser bleibt solange eingeschaltet, bis das Bit wieder zurückgesetzt, also gelöscht wird. Folglich kann bei der direkten Bedienung des Tongenerators - anders als bei der Bedienung durch den Basic-Befehl SOUND - die Tondauer nicht per Parameter gesteuert werden, sondern muß per Programmschleife programmiert werden. Der Tongenerator 1 kann ausschließlich Töne erzeugen. Anders der Tongenerator 2, der sowohl als auch Geräusche erzeugen kann. Bit 5 des Registers 17 ist für den Ton, Bit 6 desselben Registers für das Rauschen zuständig. Bei gesetztem Bit 5 (und gelöschtem Bit 6) erklingt ein Ton, bei gesetztem Bit 6 (und gelöschtem Bit 5) ein Rauschen. Setzt man beide Bits zurück, verstummt der Tongenerator 2 gänzlich. Sind auf der anderen Seite

beide Bits gesetzt, so wird ein Ton erzeugt, das heißt, der Zustand von Bit 6 wird in diesem Fall nicht beachtet. Bit 7 des Registers 17 übernimmt ebenfalls eine Schalterfunktion. Ist dieses Bit gesetzt, wird eine Sounderzeugung durch den TED - unabhängig von den übrigen Registerinhalten, also auch unabhängig von den Zuständen der Bits 4, 5 und 6 dieses Registers - gänzlich unterdrückt, das heißt, der TED bleibt völlig stumm. Ist dieses Bit gelöscht, wird die Sounderzeugung entsprechend den übrigen Parametern freigegeben. Dies ist beispielsweise dann sinnvoll, wenn man zur Voreinstellung der Parameter erst einmal beide Tongeneratoren ausgeschaltet haben will, um sie dann exakt gleichzeitig zu aktivieren.

Nun sind alle Tonregister und somit alle Grundlagen der Tonerzeugung mit dem TED geklärt, und wir können zur praktischen Erprobung des Gesagten übergehen. Geben Sie dazu bitte Listing 8 ein. REM-Zeilen können Sie dabei ersatzlos weglassen., sie dienen nur der Übersichtlichkeit des Listings und sind für den Programmablauf unrelevant. Wenn Sie das Programm nun mit RUN starten, werden im unteren Bildschirmteil die Registerinhalte der für die Tonerzeugung wichtigen Register angezeigt. Im mittleren Bildschirmteil kann man die entsprechenden Werte bzw. Zustände der einzelnen Parameter ablesen: Lautstärkewert, Zustand des 1. Tongenerators, Frequenz des 1. Tongenerators, Zustand des 2. Tongenerators und schließlich die Frequenz des 2. Tongenerators. Mit den Cursortasten können Sie diese Parameter nun beliebig verändern: Mit "oben/unten" wählen Sie den entsprechenden Parameter aus (wird revers agezeigt) und verändern ihn dann mit "rechts/links". Die Lautstärke läßt sich schrittweise von 0 bis 8 regulieren, die Frequenzen in Zehnerschritten von 0 bis 1023, und die möglichen Zustände der Tongeneratoren sind AUS, TON und beim 2. Tongenerator zusätzlich RAUSCHEN.

Probieren Sie mit diesem Programm nach Herzenslust herum und behalten Sie dabei immer die angezeigten Registerinhalte im Auge.

Dann sollten Ihnen die oben erklärten Zusammenhänge auch in der Praxis schnell klar werden.

Jetzt ziehen wir alle Register!

Einen Computer in Maschinensprache zu programmieren ist nur dann sinnvoll, wenn man eine gewisse Kenntnis des Betriebssystems und Systemspeichers des jeweiligen Gerätes besitzt. Man kann dann bequem auf computerinterne Maschinenroutinen zurückgreifen ohne jedesmal 'das Rad neu erfinden' zu müssen. Im Bezug auf den C16/116 und plus/4 sind Informationen in dieser Richtung jedoch äußerst rar, deshalb hier eine Zusammenstellung einiger wichtiger und nützlicher Betriebssystemroutinen, die sich allesamt sehr gut in eigenen Maschinenprogrammen verwenden lassen.

TIPS UND TRICKS FÜR DEN COMMODORE 16/116 UND plus/4 VON WALDEMAR RAAZ

=WICHTIGE BETRIEBSSYSTEMROUTINEN=

Adresse hex	Adresse dez	evt. Label	Beschreibung
>9491	>38033	CHKCOM	Das nächste Zeichen aus dem Basic-Text wird gelesen und auf Komma geprüft. Handelt es sich um ein anderes Zeichen, erfolgt ein SYNTAX-Error und das Programm bricht ab.
>948b	>38027		Es wird ein entsprechender Test für "Klammer zu" durchgeführt.
>948e	>38030		Es wird ein entsprechender Test für "Klammer auf" durchgeführt.
>9d84	>40324	GETBYT	Mit dieser Routine läßt sich eine sehr komfortable Parameterübergabe von Basic an Maschinensprache realisieren. Aus dem Basic-Text wird ein Ein-Byte-Zahlenwert ausgelesen. Liegt der Wert nicht in dem Bereich von 0 bis 255, so wird ein ILLEGAL QUANTITY-Error ausgegeben. Ansonsten steht der Wert im X-Register zur Verfügung.
>9de1	>40417	GETADR	Diese Routine gleicht der Routine GETBYT, nur wird hier ein Zwei-Byte-Zahlenwert ausgelesen, der sich im Bereich 0-65535 bewegen darf. Bei Bereichsunter- oder -überschreitungen erfolgt ein ILLEGAL QUANTITY-Error. Ansonsten steht der Zahlenwert nach Durchlaufen der Routine in den Speicherzellen hex.14/15 (dez.20/21) in Low-/Highbyte-Darstellung zur Verfügung.
>9dd2	>40402	GETPAR	Es werden folgende Routinen nacheinander durchlaufen: GETADR,CHKCOM,GETBYT. Das Betriebssystem benutzt diese Routine beispielsweise, um die Parameter beim POKE-Befehl auszuwerten.
>9dd8	>40408		Nacheinander werden CHKCOM und GETBYT abgearbeitet.
>9dde	>40414		Nacheinander werden CHKCOM UND GETADR abgearbeitet.
>9314	>37652	FRMNUM	Wertet einen beliebig verschachtelten oder verklammerten numerischen Ausdruck aus. Mit der Routine CHKADR (hex.9de4/dez.40420) kann dann eventuell eine Bereichsprüfung auf Zwei-Byte-Zahlenwert durchgeführt werden.
>d89a	>55450	HOME	Der Cursor wird in der linken oberen Ecke des Bildschirms positioniert. Ein Aufruf dieser Routine entspricht in der Wirkung dem Betätigen der Home-Taste im Direktmodus.
>d88b	>55435	CLRHOM	Der Bildschirm wird vollständig gelöscht und und der Cursor wird wie bei HOME in der linken oberen Bildschirmecke positioniert. Diese Routine entspricht der CLR-Taste im Direktmodus.
>daf7	>56055		Löschen einer beliebigen Bildschirmzeile. Die Nummer der zu löschenden Zeile (0-24) muß vor dem Aufruf der Routine in das X-Register geladen werden.
>88c7	>35015		Mit dieser Routine können beliebige Speicherblöcke frei verschoben werden; dazu müssen, bevor die Routine aufgerufen wird, folgende Parameter gesetzt sein: hex.5f/60 - alter Blockanfang 5a/5b - altes Blockende+1 hex.58/59 - neues Blockende+1
>903e	>36926		Durch den Aufruf dieser Routine bewirkt man die Ausgabe eines 'Carrige Return' gefolgt von einem 'Linefeed' auf dem gerade aktivierten Ausgabegerät (normalerweise ist die der Bildschirm). Man schließt auf diese Weise eine Ausgab ab und gelangt in die nächste Zeile.
>bdcd	>42079		Diese Routine gibt eine 16-Bit-Integerzahl ohne Vorzeichen auf dem Bildschirm aus. Der Zahlenwert wird dazu im X-Register/Akku in der Low-/Highbyte-Darstellung übergeben.
>9c48	>40008	FRESTR	Aus dem Basic-Text wird ein String ausgelesen und anschließend ausgewertet. Die Routine nach erfolgreichem Ablauf folgende Parameter zur Verfügung: Akku - Stringlänge X-Register - Adresse/Low-Byte Y-Register - Adresse/High-Byte
>9088	>37000		Diese Routine gibt einen String auf dem Bildschirm aus. Dazu muß der String irgendwo im Speicher im ASCII-Format stehen und mit einem Nullbyte abschließen. Außerdem müssen der Routine vor dem Aufruf folgende Parameter übergeben werden: Akku - Anfangsadresse des Strings/Low-Byte Y-Reg. - Anfangsadresse des Strings/High-Byte

			Handelt es sich bei der Ausgabe um einen String, der mit der Routine bei hex.9c48 ausgewertet wurde, so gilt folgende Einsprungadresse: hex.908b - dez.37003.
>8818	>34840	LINK	Diese Routine bindet die vorhandenen Basic-Programmzeilen neu. Alle Linkadressen werden neu berechnet und gesetzt. Nach Ablauf der Routine steht in den Speicherzellen hex.22/23 die neue Endadresse des Basic-Programms.
>884b	>34891		Nach dem Binden der Programmzeilen paßt diese Routine den Zeiger auf das Basic-Ende bzw. den Variablenanfang neu.
>8a93	>35475	CLR	Entsprechend gleicht diese Routine die übrigen Basic-Pointer an die veränderte Speicheraufteilung an.
>8703	>34563	READY	Über diese Adresse gelangt man in den READY-Modus. Hier muß man natürlich mit einem JMP-aufrufen, da es sich hier nicht um eine Unterroutine handelt. Ein solcher Einsprung ist z.B. dann sinnvoll, wenn man das C16-Basic etwa um einen OLD - Befehl erweitert, da bei diesem Befehl nicht zurückverzweigt wird.
>a86b	>43115	IO-PAR	Diese Routine liest aus dem laufenden Basictext die Parameter für LOAD, SAVE oder VERIFY aus (Prg.name,Geräteadr.,Sekundäradr.) und und besetzt die entsprechenden Speicherzellen.
>94a1	>38049		Diese Routine gibt einen SYNTAX-Error aus.
>991c	>39196		Ausgabe : ILLEGAL QUANTITY - Error
>9324	>37668		Ausgabe : TYPE MISMATCH - Error
>9327	>37671		Ausgabe FORMULA TOO COMPLEX - Error
>8683	>34435	ERROR	Diese Routine dient zur allgemeinen Fehlerbehandlung; bei Aufruf der Routine muß im X-Register die Fehlernummer (s.Handbuch S.204) übergeben werden.

BESCHREIBUNG DER TED-REGISTER BEIM COMMODORE 16/116 UND plus/4 VON WALDEMAR RAAZ

= TED - REGISTER =

Register	Adresse hex	Adresse dez	Beschreibung
00	>ff00	65280	Low-Byte des Timers #1
01	>ff01	65281	High-Byte des Timers #1
02	>ff02	65282	Low-Byte des Timers #2
03	>ff03	65283	High-Byte des Timers #2
04	>ff04	65284	Low-Byte des Timers #3
05	>ff05	65285	High-Byte des Timers #3
06	>ff06	65286	Bits0-2: Smooth Scrolling des normalen Textbildschirms in vertikaler Richtung Bit3 : Schaltbit für Anzahl der Bildschirmzeilen 0 = 24 Zeilen 1 = 25 Zeilen Bit4 : Schaltbit für Bildschirm an/aus 0 = Bildschirm 'aus' 1 = Bildschirm 'an' Bit5 : Schaltbit für hochauflösenden Grafikmodus 0 = Hochauflösungsmodus aus 1 = Hochauflösungsmodus ein Bit6 : Schaltbit f. erweiterten Hintergrundfarbmodus 0 = ausgeschaltet 1 = eingeschaltet Bit7 : Dieses Bit muß beim Betrieb auf Null gesetzt sein, ansonsten erscheinen auf dem Bildschirm seltsame Muster.
07	>ff07	65287	Bits0-2: Smooth Scrolling des normalen Textbildschirms in horizontaler Richtung Bit3 : Schaltbit für Anzahl der Bildschirmspalten 0 = 38 Spalten 1 = 40 Spalten Bit4 : Schaltbit für Mehrfarbmodus 0 = Mehrfarbmodus 'aus' 1 = Mehrfarbmodus 'an' Bit5 : Dieses Bit muß beim Betrieb auf Null gesetzt sein, da sich der Computer sonst verabschiedet und erst durch einen Reset wieder zum Leben erweckt wird. Bit6 Dieses Bit muß beim Betrieb ebenfalls auf Null gesetzt sein. Eine Eins in dieser Zelle schaltet nämlich auf NTSC-Fernsehnorm um. Eine Null steht für die heimische PAL-Norm. Bit7 . Dieses Bit bestimmt, ob ein Zeichen auf dem Bildschirm normal oder revers ausgegeben wird. Beim C16 werden reverse Zeichen anders als etwa beim C64 oder VC20 hardwaremäßig erzeugt.
08	>ff08	65288	Dieses Register wird beim C16 dazu benutzt, die Tastatur sowie die beiden Joystickports abzufragen.
09	>ff09	65289	Die Bits dieses Registers werden beim Auftreten des entsprechenden Interrupts gesetzt: Bit0 : unbenutzt Bit1 : Rasterinterrupt Bit2 : Lichtgriffelinterrupt (beim C16 nicht implementiert) Bit3 : Timer #1 ist auf Null dekrementiert Bit4 : Timer #2 ist auf Null dekrementiert Bit5 : unbenutzt Bit6 : Timer #3 ist auf Null dekrementiert Bit7 : Interrupt-Flagge - wird gesetzt, sobald ein Bit dieses Registers gesetzt wird
10	>ff0a	65290	Dieses Register ist das zu Register 09 gehörige Maskenregister. Die in 09 genannten Interruptquellen lassen sich mit diesem Register sperren bzw. freigeben. Ein ungesetztes Bit sperrt den jeweiligen Interrupt, ein gesetztes gibt ihn frei. Die Bedeutung der einzelnen Bits entspricht der Belegung bei Reg.09, bis auf Bit0, das zum nachfolgenden Register gehört.
11	>ff0b	65291	Dieses Register enthält die Nummer der Rasterzeile, in der der nächste Rasterinterrupt ausgelöst wird. Da 312 Zeilen erzeugt werden, reicht ein 8-Bit-Register zur Darstellung nicht aus, daher wird Bit0 aus Reg.10 als msb (most significant bit) hinzugezogen.
12	>ff0c	65292	Bit0 und 1 dieses Register stellen eine Ergänzung des folgenden Registers um zwei Bits dar (Bit8 und 9), womit sich insgesamt ein 10-Bit-Register ergibt.
13	>ff0d	65293	Dieses Register enthält die Bits0-7 der Position des Hardware Cursors.
14	>ff0e	65294	Bits0-7 der Frequenz des 1.Tongenerators
15	>ff0f	65295	Bits0-7 der Frequenz des 2.Tongenerators
16	>ff10	65296	Die Bits0 und 1 dieses Registers ergänzen Register15 um zwei Bits (Bits8 und 9).
17	>ff11	65297	Bits0-2: Lautstärke - gilt für beide Tongeneratoren Bit3 : nicht benutzt Bit4 : Schaltbit für den 1.Tongenerator 1 - Stimme#1 ein 0 - Stimme#1 aus Bit5 : Schaltbit für den 2.Tongenerator Inhalt wie Bit4 Bit6 : Schaltbit für den Rauschgenerator Inhalt wie Bit4 Bit7 : Beim Betrieb der Tongeneratoren muß dieses Bit gelöscht sein. Ist es gesetzt, sind alle Tongeneratoren abgeschaltet.
18	>ff12	65298	Bits0-1: Bits8 und 9 der Frequenz des 1.Tongenerators Bit2 : Schaltbit für ROM-/RAM-Bank: 0 - Daten aus RAM 1 - Daten aus ROM Bits3-5: Basisadresse der Bit-Map (Speicherraum für hochauflösende Grafik) Bits6-7: nicht benutzt
19	>ff13	65299	Bits0-1: Die Bedeutung dieser Bits habe ich noch nicht exakt klären können. Bits2-7: Basisadresse der Character-Tabelle (Zeichenmuster)
20	>ff14	65300	Bits0-2: nicht benutzt Bits3-7: Basisadresse des Bildschirmspeichers im Textmodus
21	>ff15	65301	Farbregister#1 (Hintergrundfarbe#0) Bits0-3: Farbe Bits4-6: Luminanz Bit7 : nicht benutzt
22	>ff16	65302	Farbregister#2 (Hintergrundfarbe#1) Inhalt wie Register 21

23	>ff17	65303	Farbregister#3 (Hintergrundfarbe#2) Inhalt wie Register 21
24	>ff18	65304	Farbregister#4 (Hintergrundfarbe#3) Inhalt wie Register 21
25	>ff19	65305	Farbregister#5 (Rahmenfarbe) Inhalt wie Register 21
26	>ff1a	65306	Dieses Register wird vom C16-Betriebssystem scheinbar nicht genutzt.
27	>ff1b	65307	Dieses Register wird vom C16-Betriebssystem scheinbar nicht genutzt.
28	>ff1c	65308	Das Bit0 dieses Registers ergänzt Register 29 um ein Bit (most significant bit).
29	>ff1d	65309	Dieses Register (in Verbindung mit Bit0 des vorhergehenden Registers) enthält die Nummer der Rasterzeile, die auf dem Bildschirm gerade erzeugt wird.
30	>ff1e	65310	Dieses Register enthält entsprechend die Rasterspalte; es wird vom C16-Betriebssystem aber scheinbar nicht genutzt.
31	>ff1f	65311	Dieses Register wird vom C16-Betriebssystem scheinbar nicht genutzt. Bits0-3: Blinkfrequenz Bits4-7: Diese Bits zählen die 8 Rasterzeilen, die für eine Bildschirmzeile nötig sind hoch.

Waldemar Raaz

DATAMAKER

Basiclistings von MC-Programmen schnell und komfortabel erstellt!

Bekanntlich hat der C16/116 einen Maschinensprachemonitor eingebaut, den TEDMON. Leider fehlt dem Monitor ein Befehl, mit dem man Speicherbereiche in Datazeilen ablegen kann. Dies ist immer dann notwendig, wenn man Maschinensprache und Basic kombinieren will, dann ist nämlich ein Basic-Loader nötig, der die Daten aus Datazeilen in den Speicher überträgt. Blieb einem bisher nichts anderes übrig, als die Daten von Hand in Datazeilen einzutippen, eine sehr mühsame und fehlerträchtige Arbeit, so erledigt diese Routine nun diese Aufgabe. Der vorliegende Datamaker hat gegenüber anderen noch einen entscheidenden Vorteil: er arbeitet sehr schnell. Er ist nämlich vollständig in Maschinensprache geschrieben und entsprechend schnell. Die Maschinenroutine steht letztlich im Kassettenpuffer, was folgende Konsequenzen hat:

- Daten, die im Kassettenpuffer angesiedelt sind, lassen sich mit deser Routine nicht bearbeiten. Ansonsten läßt sichder gesamte Speicherbereich in Datazeilen ablegen.
- Kassettenanwender können das Programm nicht als reine Maschinenroutine abspeichern,

```
0 REM ***********************************************************************
1 REM *                                                                     *
2 REM *       D A T A - M A K E R      FUER DEN COMMODORE 16/116            *
3 REM *                                                                     *
4 REM *       BELEGT WIRD DER SPEICHERBEREICH VOM 818 BIS 989               *
5 REM *       SYNTAX:  SYS818,AW,EW,SZ                                      *
6 REM *       ( AW-ANFANGSWERT / EW-ENDWERT / SZ-STARTZEILE )               *
7 REM *                                                                     *
8 REM *       WRITTEN  1986  BY   W A L D E M A R  R A A Z                  *
9 REM ***********************************************************************
10 :
11 :
10000 DATA 20,D0,03 :REM JSR $03D0     ; STARTWERT HOLEN
10002 DATA 8D,71,03 :REM STA $0371     ; UND ZWISCHENSPEICHERN
10004 DATA 8C,72,03 :REM STY $0372     ;
10006 DATA 20,D0,03 :REM JSR $03D0     ; ENDWERT HOLEN
10008 DATA 85,D8    :REM STA $D8       ; UND ZWISCHENSPEICHERN
10010 DATA 84,D9    :REM STY $D9       ;
10012 DATA 20,D0,03 :REM JSR $03D0     ; STARTZEILE HOLEN
10014 DATA A0,00    :REM LDY #$00      ;
10016 DATA A9,83    :REM LDA #$83      ; TOKEN FUER 'DATA' AN DEN BEGINN
10018 DATA 99,00,02 :REM STA $0200,Y   ; DES EINGABEPUFFERS SCHREIBEN
10020 DATA AD,72,03 :REM LDA $0372     ; ENDWERT SCHON ERREICHT ??
10022 DATA C5,D9    :REM CMP $D9       ;
10024 DATA 90,10    :REM BCC $0363     ; NEIN, DANN WEITERMACHEN
10026 DATA D0,07    :REM BNE $035C     ;
10028 DATA A5,D8    :REM LDA $D8       ;
10030 DATA CD,71,03 :REM CMP $0371     ;
10032 DATA B0,07    :REM BCS $0363     ;
10034 DATA C0,00    :REM CPY #$00      ; JA, TEST AUF ZEILE IM EINGABEPUFFER
10036 DATA D0,3F    :REM BNE $039F     ; POSITIV, DANN ZEILE NOCH EINBINDEN
10038 DATA 4C,03,87 :REM JMP $8703     ; NEGATIV, DANN IN DEN READY-MODUS
10040 DATA A5,15    :REM LDA $15       ; ZEILENNUMMER NOCH IM ERLAUBTEN
10042 DATA C9,FA    :REM CMP #$FA      ; BEREICH ??
10044 DATA 90,03    :REM BCC $036C     ;
10046 DATA 4C,1C,99 :REM JMP $991C     ; NEIN, DANN 'ILLEGAL QUANTITY ERROR'
10048 DATA 84,DA    :REM STY $DA       ;
10050 DATA A9,00    :REM LDA #$00      ;
10052 DATA AE,00,00 :REM LDX $0000     ; AKTUELLES BYTE HOLEN
10054 DATA 85,62    :REM STA $62       ; UND ZWISCHENSPEICHERN
10056 DATA 86,63    :REM STX $63       ;
10058 DATA A2,90    :REM LDX #$90      ;
10060 DATA 38       :REM SEC           ;
10062 DATA 20,CE,A2 :REM JSR $A2CE     ; --> FAC MIT NULLEN AUFFUELLEN
10064 DATA 20,71,A4 :REM JSR $A471     ; --> FAC NACH ASCII WANDELN
10066 DATA A4,DA    :REM LDY $DA       ;
```

da dazu der Kassettenpuffer benötigt würde. Sie müssen jedesmal, bevor sie das Programm benutzen, den Basic-Loader laden und starten.

Diskettenanwender dagegen haben die Möglichkeit, das Programm als reine Maschinenroutine auf Diskette zu speichern; es belegt in der Form nur einen Block, was sich entsprechend angenehm auf die Ladezeit auswirkt. Gehen Sie also wie folgt vor:

- Geben Sie den abgedruckten Basic-Loader ein und speichern sie ihn zur Sicherheit ab. Alle REM-Anweisungen und Doppelpunktzeilen können Sie dabei weglassen; sie sind für den Programmablauf nicht relevant und dienen nur der Übersichtlichkeit.
- Starten Sie das Programm mit RUN.
- Es dauert einen Augenblick, bis die Data-Werte eingelesen sind. Dann haben Sie entweder Pech gehabt und Ihnen ist in den Datazeilen ein Tippfehler unterlaufen, das Programm gibt dann die entsprechende Meldung aus und stoppt, oder aber Sie haben keinen Fehler begangen und alles richtig eingetippt; dann erscheint die Frage "Abspeichern (j/n)?".
- Kassettenanwender müssen hier - wie gesagt - leider ein "N" für nein eingeben. Diskettenanwender legen eine Diskette ein und drücken "J" für ja. Wenn alles in Ordnung ist, wird die Routine nun als reines Maschinenprogramm unter dem Namen "DATAGEN.CODE" auf Diskette abgespeichert. Es ist in der Form sehr kurz und belegt nur einen Block.

Für die spätere Anwendung gehen Sie bitte wie folgt vor:

- Kassettenanwender laden und starten den Basic-Loader. Diskettenanwender laden die Maschinenrroutine absolut ein:
  LOAD"DATAGEN.CODE",8,1.
  Danach geben Sie NEW ein, um den Basic-Loader zu löschen bzw. die Basicpointer wieder zu normalisieren.
  Starten Sie die Maschinenroutine mit folgender Syntax:
  SYS818,AW,EW,SZ.

```
10068 DATA A2,00     :REM LDX #$00      ; AKTUELLES BYTE IN DEZIMALER FORM
10070 DATA E8        :REM INX           ; IN DEN EINGABEPUFFER SCHREIBEN
10072 DATA C8        :REM INY           ;
10074 DATA BD,FF,00  :REM LDA $00FF,X   ;
10076 DATA 99,00,02  :REM STA $0200,Y   ;
10078 DATA D0,F6     :REM BNE $0384     ;
10080 DATA A9,2C     :REM LDA #$2C      ; MIT KOMMA ABSCHLIESSEN
10082 DATA 99,00,02  :REM STA $0200,Y   ;
10084 DATA EE,71,03  :REM INC $0371     ; AKTUELLE BYTEADRESSE ERHOEHEN
10086 DATA D0,03     :REM BNE $039B     ;
10088 DATA EE,72,03  :REM INC $0372     ;
10090 DATA C0,41     :REM CPY #$41      ; EINGABEPUFFER SCHON VOLL ??
10092 DATA 90,AD     :REM BCC $034C     ; NEIN, DANN NAECHSTES BYTE AUSWERTEN
10094 DATA A9,00     :REM LDA #$00      ; JA, DANN EINGABEPUFFER MIT
10096 DATA 99,00,02  :REM STA $0200,Y   ; EINER NULL ABSCHLIESSEN
10098 DATA C8        :REM INY           ; UND LAENGE DER ZEILE BERECHNEN
10100 DATA A9,BD     :REM LDA #$BD      ; BASIC-WARMSTART-VEKTOR UMBIEGEN
10102 DATA A2,03     :REM LDX #$03      ;
10104 DATA 8D,02,03  :REM STA $0302     ;
10106 DATA 8E,03,03  :REM STX $0303     ;
10108 DATA A2,02     :REM LDX #$02      ; ZEIGER INNERHALB DER CHRGET-ROUTINE
10110 DATA 86,3C     :REM STX $3C       ; AUF EINGABEPUFFER SETZEN
10112 DATA A2,00     :REM LDX #$00      ;
10114 DATA 86,3B     :REM STX $3B       ;
10116 DATA CA        :REM DEX           ; DIREKTMODUS-FLAG SETZEN
10118 DATA 86,3A     :REM STX $3A       ;
10120 DATA 4C,34,87  :REM JMP $8734     ; --> POGRAMMZEILE EINFUEGEN
10122 DATA A9,12     :REM LDA #$12      ; BASIC-WARMSTART-VEKTOR WIEDER NORMAL
10124 DATA A2,87     :REM LDX #$87      ;
10126 DATA 8D,02,03  :REM STA $0302     ;
10128 DATA 8E,03,03  :REM STX $0303     ;
10130 DATA E6,14     :REM INC $14       ; AKTUELLE ZEILENNUMMER ERHOEHEN
10132 DATA D0,02     :REM BNE $03CD     ;
10134 DATA E6,15     :REM INC $15       ;
10136 DATA 4C,45,03  :REM JMP $0345     ; NAECHSTES BYTE AUSWERTEN
10138 DATA 20,91,94  :REM JSR $9491     ; --> CHKCOM
10140 DATA 20,14,93  :REM JSR $9314     ; --> FRMNUM
10142 DATA 20,E4,9D  :REM JSR $9DE4     ; --> GETADR
10144 DATA A5,14     :REM LDA $14       ;
10146 DATA A4,15     :REM LDY $15       ;
10148 DATA 60        :REM RTS           ;
10200 :
10210 :   RESTORE : P=0
10220 :   FOR I =818 TO 989
10230 :     READ A$
10240 :     A = DEC(A$)
10250 :     P = P + A
10260 :     POKE I , A
10270 :   NEXT I
10280 :
10290 :   IF P = 18112 THEN 10400
10299 :
10300 PRINT"■ IHNEN IST BEIM ABTIPPEN DER DATAZEILEN "
10305 PRINT" EIN FEHLER UNTERLAUFEN.   UEBERPRUEFEN "
10310 PRINT" SIE BITTE DIE ENTSPRECHENDEN PROGRAMM- "
10315 PRINT" ZEILEN !":END
10399 :
10400 PRINT"■ DIE DATAWERTE SIND RICHTIG EINGELESEN. "
10405 PRINT" WENN SIE MIT DISKETTE ARBEITEN,KOENNEN "
10410 PRINT" SIE DAS PROGRAMM ALS REINES MASCHINEN- "
10415 PRINT" PROGRAMM ABSPEICHERN (ES BELEGT IN DER "
10420 PRINT" FORM NUR EINEN BLOCK). FUER KASSETTEN- "
10425 PRINT" ANWENDER BESTEHT DIESE OPTION NICHT,DA "
10430 PRINT" DER KASSETTENPUFFER BENOETIGT WIRD!    "
10435 PRINT" ABSPEICHERN (J/N)?"
10440 GETG$:IFG$=""THEN10440
10445 IFG$<>"J"THEN10485
10470 SYS43115 "DATAGEN.CODE",8,1
10475 POKE38,50:POKE39,3:POKE2035,222
10480 POKE2036,3:POKE2034,38:SYS65496
10485 PRINT"■ GESTARTET WIRD DURCH:"
10490 PRINT"■ SYS818,ANFANGSWERT,ENDWERT,STARTZEILE"
10495 END
10499 :
10500 REM *** PROGRAMMENDE ***

READY.
```

AW und EW bedeuten dabei Anfangs-bzw. Endwert des Speicherbereiches, der in Datazeilen abgelegt werden soll; beide Angaben haben dabei in dezimaler Form zu erfolgen und müssen sich im Bereich von 0 bis 65535 bewegen. Bei Bereichsunter- oder -überschreitung erfolgt ein ILLEGAL QUANTITY ERROR. SZ gibt die Nummer der Basic-Zeile an, bei der die Datawerte beginnen sollen. Passen nicht alle Datawerte in eine Zeile, so wird

automatisch in der nächstgrößeren Zeile fortgefahren. Eventuell schon vorhandene Basiczeilen gleicher Nummer werden dabei gelöscht; die anderen bleiben unberührt. Bedingt durch die Speicherorganisation des C16/116 ist die höchst erlaubte Zeilennummer 63999. Wird SZ ein höherer Wert zugewiesen oder im Ablauf der Routine erreicht, bricht der Programmablauf ab und es erfolgt ein ILLEGAL QUANTITY ERROR.

- Nach dem Ablauf der Routine (bei großen Datenmengen kann dies einen Moment dauern - zaubern kann man auch mit Maschinensprache nicht) stehen die entsprechenden Datazeilen im Speicher. Alle Basicpointer sind richtig eingestellt, so daß die Datazeilen um einen Einleseteil ergänzt und abgespeichert werden können. Es lassen sich also sehr komfortabel sowohl Basic-Loader für Maschinenroutinen als auch Datentabellen z. B. für einen eigenen Zeichensatz erstellen.

Das Programm greift auf eine Reihe sehr interessanter Betriebssystemroutinen des C16/116 zurück. Ich habe daher in den abgedruckten Basic-Loader ein sehr ausführlich dokumentiertes Assemblerlisting der Routine integriert, das für Maschinensprache-programmierer sehr interessant sein dürfte.

Waldemar Raaz

AUTO-OLD-ROUTINE
für den Commodore 16/116

Bekanntlich hat auch der C16/116 trotz seines komfortablen Basics einige OGNV-Befehle (oft gebraucht, nie vorhanden), zu denen sicherlich auch der OLD- oder RENEW-Befehl, wie er in anderen Basic-Versionen genannt wird, gehört. Hat man nämlich ein Basic-Programm versehentlich mit dem Befehl "NEW" oder durch einen RESET gelöscht, kann man es mit dem OLD-Befehl wieder herstellen. Dies ist möglich, da ein gelöschtes Basic-Programm nicht etwa überschrieben wird, es werden vielmehr nur alle Pointer so eingestellt, daß es für den Computer den Anschein hat, als sei der Basic-Speicher leer. Nun gibt es mittlerweile auch für den C16/116 schon eine Reihe von diversen OLD-Routinen, doch hat die vorliegende Routine - meiner Meinung nach - gegenüber anderen einen entscheidenden Vorteil. Muß man sich bei herkömmlichen OLD-Routinen immer die Startadresse merken, da solch eine Routine zwangsläufig in Maschinensprache geschrieben sein muß und folglich durch SYS gestartet wird, so ist dies bei der abgedruckten Routine nicht mehr nötig. Sie ist mit einem Autostarter versehen, der die Routine nach dem Laden (sowohl (!) von Diskette als auch von Kassette) automatisch startet. Das mehr als lästige Suchen der Startadresse entfällt somit. Gehen Sie zum Eingeben der Routine folgendermaßen vor:

- geben Sie den abgedruckten Basic-Loader ein und speichern Sie ihn zur Sicherheit ab. Alle REM-Anweisungen und Doppelpunktzeilen können Sie dabei weglassen; sie sind für den Programmablauf nicht relevant.
- Starten Sie den Basic-Loader mit RUN.
- Es dauert einen Augenblick, bis die DATA-Werte eingelesen sind. Das Programm führt beim Einlesen einen zeilenorientierten Prüfsummencheck durch. Das heißt, fehlerhafte DATA-Zeilen werden sofort lokalisiert. Das Programm gibt Ihnen dann die Nummer der Zeile an und stoppt.
- Geben Sie auf die Frage nach dem Speichermedium entsprechend Ihres Speichers "D" oder "K" ein. Legen Sie dann eine Diskette/Kassette ein und drücken die "SPACE"-Taste.
- Wenn alles in Ordnung ist, wird die Routine jetzt automatisch abgespeichert. Sie ist extrem kurz, was sich angenehm in den Speicher- und Ladezeiten niederschlägt.

Haben Sie nun versehentlich ein Basic-Programm gelöscht, nehmen Sie die generierte Maschinenroutine (nicht etwa den Basic-Loader) und laden ihn sofort nach dem Löschen ein. Sind nach dem Löschen Variablen angelegt oder Basiczeilen eingegeben worden, läßt sich das gelöschte Programm nicht mehr regenerieren. Zum Laden geben Diskettenanwender

LOAD"AUTO-OLD.CODE",8,1

und Kassettenanwender entsprechend

LOAD"AUTO-OLD.CODE",1,1

ein. Nach dem Laden startet die Routine automatisch und regeneriert das gelöschte Basic-Programm. Dabei erscheint sowohl nach dem Laden als auch nach Ablauf der Routine ein READY. auf dem Bildschirm, das ist normal. Mit List können Sie sich nun überzeugen, ob alles einwandfrei geklappt hat. Da für Maschinenprogrammierer sowohl die verwendeten Betriebssystemroutinen als auch die Technik des Autostartes sehr interessant sein dürfte, habe ich dem Basic-Loader ein sehr ausführliches dokumentiertes Assemblerlisting der Routine beigelegt.

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

0 REM ******************************************
*****************************                    <171>
1 REM *
                                 *               <98>
2 REM *       A U T O - O L D  R O U T I N E  FU
ER DEN COMMODORE 16/116          *               <118>
3 REM *
                                 *               <100>
4 REM *       BELEGT WIRD DER SPEICHERBEREICH VO
M 709 BIS 771                    *               <112>
5 REM *
                                 *               <102>
6 REM *       DER WARMSTARTVEKTOR ($0302/0303) W
IRD MODIFIZIERT                  *               <25>
7 REM *
                                 *               <104>
8 REM *       WRITTEN  1986  BY   W A L D E M A
R  R A A Z                       *               <228>
9 REM ******************************************
*****************************                    <180>
10 :                                             <68>
11 :                                             <69>
10000 DATAA2,12,A0,87,8E,02,03,8C, 762           <101>
10002 DATA03,03,A9,01,A8,91,2B,20, 564           <81>
10004 DATA1B,88,20,4B,88,20,93,8A, 720           <157>
10006 DATA4C,03,87,00,00,00,00,D0, 422           <84>
10008 DATA00,00,00,00,15,00,00,00, 21            <74>
10010 DATA00,00,00,00,71,98,71, 378              <109>
10012 DATA94,00,00,00,00,00,00,00, 148           <106>
10014 DATA00,00,00,86,86,C5,02,00, 467           <231>
10016 DATAA2,3E,BD,00,30,9D,C5,02, 817           <210>
10018 DATACA,10,F7,A2,C5,A0,02,86, 1120          <209>
10020 DATA26,84,27,A2,08,A0,FF,20, 826           <97>
10022 DATABA,FF,A9,0D,A2,77,A0,30, 1112          <151>
10024 DATA20,BD,FF,A9,26,A2,04,A0, 1009          <138>
10026 DATA03,20,D8,FF,A2,12,A0,87, 981           <188>
10028 DATABE,02,03,8C,03,03,60,41, 454           <86>
10030 DATA55,54,4F,2D,4F,4C,44,2E, 562           <112>
10032 DATA43,4F,44,45,00,00,00,00, 283           <175>
10100 :                                          <213>
10110 :    RESTORE                               <247>
10120 :    FOR T = 12288 TO 12420 STEP 8         <57>
10130 :      P = 0                               <105>
10140 :      FOR I = 0 TO 7                      <235>
10150 :        READ A$                           <104>
10160 :        A = DEC(A$)                       <127>
10170 :        P = P + A                         <233>
10180 :        POKE T+I , A                      <51>
10190 :      NEXT I                              <14>
10200 :    READ R                                <61>
10210 :    IF P <> R THEN 10300                  <141>
10220 :    NEXT T                                <77>
10230 :    GOTO 10400                            <48>
10240 :                                          <98>
10250 :                                          <108>
10300 PRINT"{CLEAR DOWN SPACE}IHNEN{SPACE}IST{SP
ACE}BEIM{SPACE}ABTIPPEN{SPACE}DER{SPACE}DATAZEIL
EN{SPACE}"                                       <238>
```

```
10310 PRINT"{DOWN SPACE}EIN{SPACE}FEHLER{SPACE}U
NTERLAUFEN.{SPACE}IN{SPACE}ZEILE{RVSON}"PEEK(63)
+PEEK(64)*256                                   <233>
10320 PRINT"{SPACE}IST{SPACE}EIN{SPACE}PRUEFSUMM
ENFEHLER{SPACE}AUFGETAUCHT!{SPACE}"             <146>
10330 PRINT"{DOWN SPACE}BITTE{SPACE}UEBEPRUEFEN{
SPACE}SIE{SPACE}DIESE{SPACE}ZEILE{SPACE}!":END  <218>
10340 :                                         <198>
10350 :                                         <208>
10400 PRINT"{CLEAR DOWN SPACE}DIE{SPACE}DATAWERT
E{SPACE}SIND{SPACE}RICHTIG{SPACE}EINGELESEN.{SPA
CE}"                                            <227>
10410 PRINT"{DOWN SPACE}DIE{SPACE}AUTO-OLD{SPACE
}ROUTINE{SPACE}WIRD{SPACE}NUN{SPACE}ALS{SPACE}RE
I-{SPACE}"                                      <234>
10420 PRINT"{DOWN SPACE}NES{SPACE2}MASCHINENPROG
RAMM{SPACE2}ABGESPEICHERT.{SPACE}"              <17>
10430 PRINT"{DOWN SPACE}ARBEITEN{SPACE}SIE{SPACE
}MIT{SPACE2 RVSON}D{RVSOFF}ISK{SPACE}ODER{SPACE2
 RVSON}K{RVSOFF}ASSETTE?{SPACE}"                <16>
10440 GETKEYG$:IFG$<>"D"ANDG$<>"K"THEN10440     <27>
10450 GA=8:IFG$="K"THENGA=1                     <34>
10460 POKE12372,GA:PRINT"{DOWN SPACE RVSON}OK!{R
VSOFF}"                                         <40>
10470 PRINT"{DOWN SPACE}LEGEN{SPACE}SIE{SPACE}JE
TZT{SPACE}EINE{SPACE}DISKETTE/KASSETTE{SPACE}"  <74>
10480 PRINT"{DOWN SPACE}EIN{SPACE}UND{SPACE}DRUE
CKEN{SPACE}SIE{SPACE}DIE{SPACE RVSON}SPACE{RVSOF
F}-TASTE{SPACE}!{SPACE}"                        <246>
10490 GETG$:IFG$<>"{SPACE}"THEN10490            <129>
10500 PRINT"{DOWN SPACE RVSON}OK!{RVSOFF}":SYS12
352                                             <58>
10510 PRINT"{CLEAR DOWN SPACE}DIE{SPACE}AUTOOLD{
SPACE}ROUTINE{SPACE}IST{SPACE}ABGESPEICHERT!{SPA
CE}"                                            <23>
10520 PRINT"{DOWN SPACE}ZUR{SPACE}ANWENDUNG{SPAC
E2}LAEDT{SPACE}MAN{SPACE}SIE{SPACE}ABSOLUT{SPACE
}!{SPACE}"                                      <78>
10530 PRINT"{DOWN SPACE}SIE{SPACE}STARTET{SPACE}
DANN{SPACE}AUTOMATISCH{SPACE}!" : END           <16>
10540 :                                         <143>
10550 REM *** PROGRAMMENDE ***                  <137>
```

ENDE DES LISTINGS

Sehr praktisch: 2 Bildschirme

DOUBLE SCREEN ist ein Maschinensprachprogramm, das die Verwaltung von zwei Textbildschirmen ermöglicht.

Das Programm ist in die Basic-Zeile 0 eingelagert und kann so sehr einfach geladen, gespeichert und in Basic-Programme eingebaut werden.

Beim Listen der Zeile 0 wird automatisch der Bildschirm gelöscht und die Zeile 0 nicht angezeigt, da ein RETURN auf der Zeile die Zerstörung des Programmes bedeuten würde. Will man das Unterprogramm aufrufen, muß zuerst der Graphic-Speicher eingerichtet werden.

GRAPHIC 1,1: GRAPHIC 0 oder SYS 50748

Nähere Informationen über das Programm befinden sich im zweiten Programm, einem Demo-Programm, unter den Punkten 4 und 5. Es sei hier nur kurz darauf hingewiesen, daß das Umschalten der Bildschirme durch programmieren eines TED-Registers erfolgt. Das Demo-Programm zeigt drei Möglichkeiten der Anwendung, die allerdings wenig praktischen Wert haben und nur zur Demonstration dienen. Es sind trotzdem erstaunliche Effekte.

Unter Punkt 4 des Demo-Menüs werden die einzelnen Befehle erklärt, die die Unterroutine bietet. Unter Punkt 5 sind dann Hinweise, die sich auf das Maschinen-Programm und auf das Arbeiten mit dem Prrogramm beziehen. Auf der beiliegenden Diskette (mit einer 1551 abgespeichert) befinden sich nun zwei Programme:

1. **DOUBLE SCREEN (L)**
   ist der DATA-Loader für das Maschinen-Programm.
2. **DS-DEMO (ohne 0)**
   ist das Demo ohne die Zeile 0.

Zuerst sollte man den DATA-Loader eintippen und ihn aufrufen. Darauf aufbauend kann dann das DEMO-Programm eingegeben werden.

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
100 REM- DOUBLESCREEN                           <214>
110 REM-                                        <87>
120 REM- (C) 1986                               <13>
130 REM- BY INSWAB                              <176>
140 :                                           <198>
150 :                                           <208>
160 SYS 50748                                   <104>
170 :                                           <228>
180 AD=30000                                    <63>
190 :                                           <248>
200 FOR Z=1 TO 22                               <149>
210 :                                           <12>
220 :S=0                                        <146>
230 :FOR I=1 TO 8                               <26>
240 ::READ H$                                   <7>
250 ::POKE AD,DEC(H$)                           <119>
260 ::AD=AD+1                                   <107>
270 ::S=S+DEC(H$)                               <195>
280 :NEXT                                       <87>
290 :                                           <93>
300 :READ S$                                    <254>
310 :IF S$=RIGHT$(HEX$(S),2) THEN 360           <24>
320 :PRINT "PRUEFSUMMENFEHLER{SPACE}IN{SPACE}ZEI
LE";                                            <165>
330 :PRINT Z*10+570                             <167>
340 :END                                        <143>
350 :                                           <153>
360 NEXT                                        <235>
370 :                                           <173>
380 C$=CHR$(13)                                 <213>
390 D$=CHR$(17)                                 <252>
400 E$=CHR$(27)                                 <13>
410 Q$=CHR$(34)                                 <20>
420 R$=CHR$(141)                                <196>
430 L$=CHR$(157)                                <255>
440 :                                           <243>
450 K$=E$+"N"+D$+L$+E$+"B{SPACE}MONITOR"+C$     <206>
460 K$=K$+"T{SPACE}7530{SPACE}7630{SPACE}4000"+C
$+"X"+C$                                        <28>
470 K$=K$+"KEY{SPACE}1,"+Q$+"GRAPHIC"+C$        <233>
480 K$=K$+"CLR"+C$+"DELETE{SPACE}1"+C$          <118>
490 K$=K$+E$+"NOK."+R$                          <186>
500 :                                           <47>
510 KEY 1,K$                                    <190>
520 POKE 1373,68                                <185>
530 POKE 1374,0                                 <222>
540 END                                         <158>
550 :                                           <98>
560 :                                           <108>
570 :                                           <118>
580 DATA 00,AD,40,00,00,8F,1B,4E,E5             <53>
590 DATA 1B,42,1B,54,20,20,22,4C,7A             <81>
600 DATA A1,94,C9,53,D0,F9,20,73,AD             <130>
610 DATA 04,A2,08,C9,31,F0,06,A2,40             <5>
620 DATA 10,C9,32,D0,0E,86,61,20,F0             <148>
630 DATA 73,04,AD,14,FF,29,07,05,6C             <107>
640 DATA 61,D0,05,AD,14,FF,49,18,57             <159>
650 DATA A8,8C,14,FF,20,79,04,C9,AD             <213>
660 DATA 2C,F0,0A,A0,FF,C8,8C,03,1C             <219>
670 DATA 40,8C,04,40,60,20,73,04,07             <83>
680 DATA C9,54,D0,BE,20,73,04,A2,E4             <67>
690 DATA 24,C9,31,F0,0C,C9,32,D0,E5             <172>
700 DATA 11,8E,84,40,8E,86,40,F0,A7             <188>
710 DATA 06,8E,81,40,8E,89,40,20,CC             <154>
720 DATA 73,04,A9,08,A2,10,A0,FF,79             <193>
730 DATA C8,84,61,85,62,84,63,86,01             <230>
740 DATA 64,B1,61,AA,B1,63,91,61,26             <105>
750 DATA 8A,91,63,C8,D0,F3,E6,62,51             <130>
760 DATA E6,64,A5,62,C9,10,D0,E9,E3             <62>
770 DATA A0,B1,A2,91,8C,81,40,8C,5D             <136>
780 DATA 84,40,8E,86,40,8E,89,40,6F             <32>
790 DATA D0,92,1B,4E,00,00,00,00,CB             <131>
```

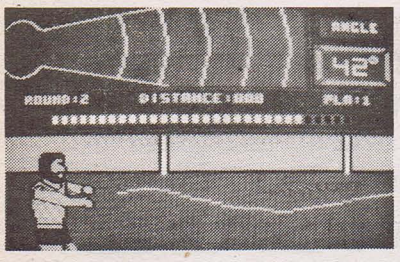
ENDE DES LISTINGS

Versandhandel R. Lindenschmidt

* * *

are proudly to present

* * * * * * * * * *



European Games

5 neue Sportarten für den C-16

- * Hammerwerfen (siehe Foto)
- * Schwimmen
- * Rudern
- * Gewichtheben
- * Weitsprung

DM 19,95

Diese Spiele sind einfach Spitze

Torpedo Run	Cass.	18,95
Speedking	Cass.	9,95
Superhits	Cass.	27,95
Favourite Four	Cass.	26,95
Project Nova	Cass.	24,95
Winter Olympiade	Cass.	29,95
Kikstart	Cass.	9,95

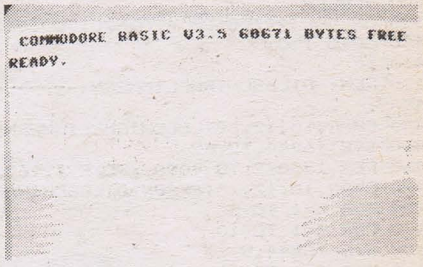
Neuheiten:

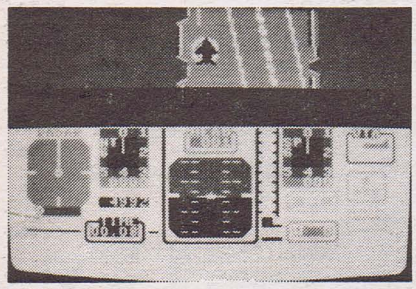
Mount Vesuvius	Cass.	26,95
Suicide Run	Cass.	9,95
Winnie the Witch	Cass.	9,95
Trailblazer	Cass.	25,95
Classics III	Cass.	33,95
Kane	Cass.	9,95
Netrun	Cass.	9,95
Terra Cognita	Cass.	9,95
Ghost'n Ghoblins	Cass.	29,95

64 k Steckmodul

Die Supererweiterung für den C-16

- * 60671 Bytes free
- * Damit ist der C-16 dem 64'er speicherplatzmäßig klar überlegen (22000 Bytes mehr als der C-64)
- * Voll vom Basic her nutzbar
- * Wahnsinnsspiele für die ausgebaute Version





Jump Jet

(mindestens 16 k Erweiterung nötig)

Senkrechtstarter (Flugsimulator)

- * Start vom Flugzeugträger aus
- * Deutsche Anleitung
- * Top Grafik mit verschiedenen Perspektiven
- * Sehr gute Instrumentierung (siehe Foto)
- * Sehr realistische Simulation

DM 33,75

Superspiele für den ausgebauten C-16

Operation Thunderstorm
Ein deutschsprachiges Textadventure, daß es in sich hat. Witzig und locker aufgebaut und nur am Anfang einfach (oder nicht?) Cass. 29,95 Disk. 29,95

ACE (64 k) Flugsimulator, der die 64 k voll ausnutzt! Einfach super! Cass. 38,95

Quiwi Das Quizspiel für die ganze Familie. Cass. 29,95

Jump Jet Flugsimulator eines Senkrechtstarters Cass. 33,75

Kurvendiskussion
Eine tolle Hilfe für Schule/Beruf. Minima, Maxima, Nullstellen, Polstellen, Wendepunkte, Lupenfunktion, Grafik ständig abrufbar, etc. Cass. 19,95 Disk. 19,95

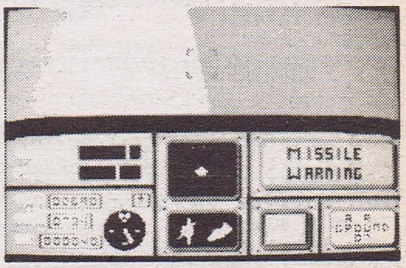
Literatur und Zubehör

Data Becker: Basic Buch			29,00
Data Becker: Grafikbuch			29,00
Data Becker: Maschinenspr.			29,00
Leercassetten	C-15	5 St.	11,95
Leercassetten	C-5	5 St.	9,95
C-16 Userport			59,95
Leerdisketten		10 St.	19,95

ACE (64 k) DM 38,95

Ein Kampfflugsimulator der Superlative

- * verschiedene Bewaffnung wählbar
- * Landkartenanzeige
- * Boden, Luft und Schiffskampf
- * Betankung in der Luft möglich
- * Spitzengrafik, wahnsinnig schnell
- * Gegnerischer Raketenangriff wird angezeigt



Unsere Lieferbedingungen

Einfach spitze: Wir liefern Ihnen die Ware frei Haus – **auch nach Österreich, Luxemburg, Belgien, Schweiz, Niederlande** (nur der Zoll geht zu Lasten des Empfängers)! Sie bezahlen keine zusätzlichen Nebenkosten, Porto oder Nachnahmegebühren. Einen Mindestbestellwert kennen wir ebenfalls nicht. Durch unseren Telefonservice (keinen Anrufbeantworter – sondern einen Ansprechpartner) können Sie problemlos bei uns bestellen. Unser umfangreiches Lager und das riesige Softwareangebot sorgen für zufriedene Kunden.

Spitzenservice + Superqualität – hierfür bürgt der Name Lindenschmidt!

Versandhandel R. Lindenschmidt · Schulstraße 14 · Postfach 1328 · Telefon (05732) 72849

Doublescreen-Demo

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
100 REM- DEMO PROGRAMM                                    <134>
110 REM- (C) 1986                                         <3>
120 REM- BY INSWAB                                        <166>
130 :                                                     <188>
140 SYS 50748                                             <84>
150 SCNCLR                                                <126>
160 SYS 16444,T2,S2                                       <133>
170 PRINT TAB(6) CHR$(27) "T"                             <177>
180 :                                                     <238>
190 PRINT "DOUBLESCREEN{SPACE}DEMONSTRATION"              <10>
200 PRINT "==========================="                  <134>
210 PRINT                                                 <107>
220 PRINT                                                 <117>
230 PRINT "BILDSCHIRMFLIMMERN.......1                     <32>
240 PRINT                                                 <137>
250 PRINT "LAUFBAENDER..............2                     <183>
260 PRINT                                                 <158>
270 PRINT "ZENTRALBEWEGUNG..........3                     <20>
280 PRINT                                                 <178>
290 PRINT "BEFEHLSERLAEUTERUNG......4                     <220>
300 PRINT                                                 <198>
310 PRINT "HINWEISE.................5                     <62>
320 PRINT CHR$(19)CHR$(19)                                <30>
330 :                                                     <133>
340 SYS 16444,S1                                          <136>
350 :                                                     <153>
360 GETKEY A$                                             <79>
370 IF A$<"1" OR A$>"5" THEN 360                          <86>
380 SCNCLR                                                <101>
390 ON VAL(A$)GOSUB 420,570,750,910,1380                  <189>
400 GOTO 150                                              <219>
410 :                                                     <213>
420 REM--BILDSCHIRMFLIMMERN-----------                    <86>
430 :                                                     <233>
440 CHAR,10,12,"BILDSCHIRMFLIMMERN"                       <92>
450 SYS 16444,T2,S2                                       <168>
460 FOR I=3072 TO 4090 :POKE I,160 :NEXT                  <235>
470 CHAR,10,12, "{RVSON}BILDSCHIRMFLIMMERN"               <139>
480 FOR K=1 TO 5                                          <68>
490 FOR I=1 TO 10                                         <130>
500 SYS 16444,S                                           <111>
510 FOR J=1 TO 60 :NEXT J,I                               <76>
520 FOR I=1 TO 50                                         <185>
530 SYS 16444,S                                           <142>
540 FOR J=1 TO 5  :NEXTJ,I,K                              <121>
550 RETURN                                                <182>
560 :                                                     <108>
570 REM--LAUFBAENDER------------------                    <246>
580 :                                                     <128>
590 SYS 16444,T2,S2                                       <53>
600 FOR I=1 TO 20: PRINT"{SPACE RVSON SPACE RVSO
FF}";: NEXT                                               <252>
610 FOR I=1 TO 5: PRINT"{RVSON SPACE}"SPC(38)"{S
PACE}"                                                    <23>
620 PRINT:NEXT                                            <1>
630 FOR I=1 TO 20: PRINT"{SPACE RVSON SPACE RVSO
FF}";: NEXT                                               <26>
640 CHAR,14,5,"LAUFBAENDER"                               <143>
650 SYS 16444,T                                           <14>
660 SCNCLR                                                <126>
670 FOR I=1 TO 20: PRINT"{RVSON SPACE RVSOFF SPA
CE}";: NEXT                                               <222>
680 PRINT                                                 <67>
690 FOR I=1 TO 5: PRINT                                   <113>
700 PRINT"{RVSON SPACE}"SPC(38)"{SPACE}": NEXT            <207>
710 FOR I=1 TO 20: PRINT"{RVSON SPACE RVSOFF SPA
CE}";: NEXT                                               <6>
720 CHAR,14,5,"LAUFBAENDER"                               <223>
730 GOTO 480                                              <53>
740 :                                                     <32>
750 REM--ZENTRALBEWEGUNG---------------                   <136>
760 :                                                     <52>
770 SYS 16444,T2,S2                                       <234>
780 FOR I=1 TO 40:H$=H$+"{RVSON SPACE}":NEXT              <182>
790 FOR I=1 TO 10                                         <176>
800 CHAR,I,I,RIGHT$(H$,80-I*4)                            <181>
810 FOR J=I TO 22-I                                       <35>
820 CHAR,I,J,"{RVSON SPACE}": NEXT                        <252>
830 CHAR,I,22-I,RIGHT$(H$,80-I*4)                         <207>
840 FOR J=I TO 22-I                                       <65>
850 CHAR,39-I,J,"{RVSON SPACE}": NEXT                     <69>
860 SYS 16444,T                                           <225>
870 NEXT                                                  <235>
880 H$=""                                                 <75>
890 GOTO 480                                              <214>
900 :                                                     <193>
910 REM--BEFEHLSMOEGLICHKEITEN                            <173>
920 :                                                     <213>
930 PRINT                                                 <62>
940 PRINT TAB(9)"BEFEHLSMOEGLICHKEITEN"                   <203>
950 PRINT TAB(9)"====================="                  <55>
960 PRINT                                                 <92>
970 PRINT TAB(6) CHR$(27) "T"                             <212>
980 PRINT SPC(8)"SYS{SPACE}16444,X"                       <73>
990 PRINT                                                 <122>
1000 PRINT "{SPACE}FUER{SPACE}>X<{SPACE}KOENNEN{
SPACE}FOLGENDE"                                           <20>
1010 PRINT SPC(5)"KOMMANDOS{SPACE}STEHEN:"                <26>
1020 PRINT "----------------------------"                 <18>
1030 PRINT "S{SPACE}=>BILDSCHIRME{SPACE}UMSCHALT
EN                                                        <116>
1040 PRINT "S1=>BILDSCHIRM{SPACE}1{SPACE}EINSCHA
LTEN                                                      <69>
1050 PRINT "S2=>BILDSCHIRM{SPACE}2{SPACE}EINSCHA
LTEN                                                      <101>
1060 PRINT "T{SPACE}=>BS-INHALTE{SPACE}AUSTAUSCH
EN                                                        <114>
1070 PRINT "T1=>INHALT{SPACE}BS{SPACE}2{SPACE}IN
{SPACE}BS{SPACE}1                                         <225>
1080 PRINT "T2=>INHALT{SPACE}BS{SPACE}1{SPACE}IN
{SPACE}BS{SPACE}2                                         <247>
1090 PRINT "----------------------------"                 <89>
1100 PRINT "ES{SPACE}KOENNEN{SPACE}MEHRERE{SPACE
}KOMMANDOS                                                <100>
1110 PRINT SPC(3)"HINTEREINANDER{SPACE}STEHEN.            <198>
1120 PRINT "(Z.B.:{SPACE}SYS{SPACE}16444,S1,T,S2
...)                                                      <220>
1130 PRINT                                                <7>
1140 PRINT                                                <17>
1150 PRINT SPC(5)"WEITER{SPACE}MIT{SPACE}>SPACE<          <209>
1160 GOSUB 2130                                           <204>
1170 SCNCLR                                               <126>
1180 PRINT SPC(9)"SYS{SPACE}16444"                        <189>
1190 PRINT                                                <67>
1200 PRINT "NACH{SPACE}EINEM{SPACE}RENUMBER-BEFE
HL                                                        <212>
1210 PRINT "{SPACE}VERAENDERT{SPACE}SICH{SPACE}A
UCH{SPACE}DIE                                             <172>
1220 PRINT SPC(9)"ZEILE{SPACE}0.                          <65>
1230 PRINT                                                <107>
1240 PRINT "Z.B.:{SPACE}RENUMBER{SPACE}100,10             <17>
1250 PRINT "{SPACE6}ZEILE{SPACE}0{SPACE}WIRD{SPA
CE}ZU{SPACE}100                                           <138>
1260 PRINT                                                <137>
1270 PRINT "DURCH{SPACE}SYS{SPACE}16444{SPACE}WI
RD{SPACE}ZEILE                                            <196>
1280 PRINT SPC(5)"100{SPACE}WIEDER{SPACE}ZU{SPAC
E}0.                                                      <148>
1290 PRINT                                                <168>
1300 PRINT                                                <178>
1310 PRINT SPC(4)"INS{SPACE}MENUE{SPACE}MIT{SPAC
E}>E<                                                     <180>
1320 GET A$: IFA$="E"THEN1340                             <238>
1330 GOTO 1320                                            <115>
1340 PRINTCHR$(19)CHR$(19)                                <30>
1350 RETURN                                               <217>
1360 GOTO1360                                             <161>
1370 :                                                    <153>
1380 REM--HINWEISE-------------------                     <172>
1390 :                                                    <173>
1400 PRINT "HINWEISE:                                     <44>
1410 PRINT "=========                                     <103>
1420 PRINT                                                <42>
1430 PRINT CHR$(27)"T";                                   <25>
1440 PRINT "NACH{SPACE}DEM{SPACE}DER{SPACE}PROGR
AMMLADER{SPACE}VON                                        <123>
1450 PRINT "DOUBLESCREEN{SPACE}ABGELAUFEN{SPACE}
IST,                                                      <42>
1460 PRINT "BEFINDET{SPACE}SICH{SPACE}DOUBLESCRE
EN{SPACE}IN{SPACE}ZEILE{SPACE}0.                          <225>
1470 PRINT                                                <92>
1480 PRINT "BEIM{SPACE}LISTEN{SPACE}WIRD{SPACE}N
UN{SPACE}DER{SPACE}BILDSCHIRM                             <237>
1490 PRINT "GELOESCHT{SPACE}UND{SPACE}ZEILE{SPAC
```

```
E}O{SPACE}NICHT{SPACE}ANGEZEIGT.                                <112>
1500 PRINT                                                      <122>
1510 PRINT "ZEILE{SPACE}O{SPACE}DARF{SPACE}VON{S
PACE}NUN{SPACE}AN{SPACE}NICHT{SPACE}MEHR                        <190>
1520 PRINT "BENUTZT{SPACE}WERDEN,{SPACE}DA{SPACE
}SONST{SPACE}DOUBLESCREEN                                       <76>
1530 PRINT "VERLOREN{SPACE}GEHT!                                <167>
1540 PRINT                                                      <163>
1550 PRINT "NACH{SPACE}EINEM{SPACE}RENUMBER{SPAC
E}MUSS{SPACE}SOFORT{SPACE}SYS                                   <184>
1560 PRINT "AUFGERUFEN{SPACE}WERDEN{SPACE}(SIEHE
{SPACE}>4<)                                                     <160>
1570 PRINT                                                      <193>
1580 PRINT "DAS{SPACE}PROGRAMM{SPACE}LAEUFT{SPAC
E}NUR,{SPACE}WENN{SPACE}GRAPHIC                                 <210>
1590 PRINT "EINGESCHALTET{SPACE}IST!                            <178>
1600 PRINT                                                      <223>
1610 PRINT "SPEICHERT{SPACE}MAN{SPACE}EIN{SPACE}
PROGRAMM{SPACE}AB,{SPACE}IN{SPACE}DEM                           <90>
1620 PRINT "DOUBLESCREEN{SPACE}ALS{SPACE}ZEILE{S
PACE}O{SPACE}VORHANDEN{SPACE}IST,                               <40>
1630 PRINT "SO{SPACE}WIRD{SPACE}ZEILE{SPACE}O{SP
ACE}MIT{SPACE}ABGESPEICHERT!                                    <178>
1640 PRINT                                                      <7>
1650 PRINT "WEITER{SPACE}MIT{SPACE}>SPACE<";                    <14>
1660 PRINT "{UP}"CHR$(27)"K{LEFT}";CHR$(27)"B";                 <128>
1670 GOSUB 2130                                                 <204>
1680 SCNCLR                                                     <126>
1690 PRINT "WENN{SPACE}BILDSCHIRM{SPACE}2{SPACE}
EINGESCHALTET{SPACE}IST,                                        <183>
1700 PRINT "ERFOLGEN{SPACE}PRINT-ANWEISUNGEN{SPA
CE}TROTZDEM{SPACE}AUF{SPACE}BILDSCHIRM{SPACE}1{S
PACE}!                                                          <130>
1710 PRINT                                                      <77>
1720 PRINT "SOBALD{SPACE}EIN{SPACE}ERROR{SPACE}A
UFTRITT{SPACE}WIRD                                              <81>
1730 PRINT "AUTOMATISCH{SPACE}BILDSCHIRM{SPACE}1
{SPACE}EINGESCHALTET.                                           <101>
1740 PRINT                                                      <107>
1750 PRINT "DER{SPACE}CURSOR{SPACE}ERSCHEINT{SPA
CE}UNABHAENGIG{SPACE}VOM                                        <248>
1760 PRINT "EINGESCHALTETEN{SPACE}BILDSCHIRM.                   <192>
1770 PRINT                                                      <137>
1780 PRINT "WENN{SPACE}GRAPHIC{SPACE}EINGESCHALT
ET{SPACE}IST,                                                   <30>
1790 PRINT "ENTSTEHT{SPACE}EIN{SPACE}FUER{SPACE}
BASIC{SPACE}UNBENUTZBARES                                       <148>
1800 PRINT "LOCH{SPACE}IM{SPACE}RAM.                            <39>
1810 PRINT "IN{SPACE}DIESEM{SPACE}LOCH{SPACE}RIC
HTET{SPACE}DOUBLESCREEN{SPACE}DEN                               <69>
1820 PRINT "2.{SPACE}BILDSCHIRM{SPACE}EIN.                      <137>
1830 PRINT                                                      <198>
1840 PRINT "DIESER{SPACE}BEREICH{SPACE}IST{SPACE
}VON{SPACE}NUN{SPACE}AN{SPACE}NICHT                             <1>
1850 PRINT "MEHR{SPACE}FUER{SPACE}MASCHIENENPROG
RAMME{SPACE}ZU                                                  <193>
1860 PRINT "GEBRAUCHEN!                                         <45>
1870 GOSUB 2130                                                 <149>
1880 SCNCLR                                                     <71>
1890 PRINT "DOUBLESCREEN{SPACE}IST{SPACE}NUR{SPA
CE}LAUFFAEHIG,                                                  <118>
1900 PRINT "WENN{SPACE}DER{SPACE}BASIC-RAM{SPACE
}BEREICH{SPACE}BEI{SPACE}$4000                                  <180>
1910 PRINT "BEGINNT.                                            <159>
1920 PRINT                                                      <32>
1930 PRINT "SETZT{SPACE}MAN{SPACE}DEN{SPACE}BASI
C-ANFANG{SPACE}HOCH,{SPACE}SO                                   <199>
1940 PRINT "MUSS{SPACE}DAS{SPACE}PROGRAMM{SPACE}
ANGEPASST{SPACE}WERDEN.                                         <215>
1950 PRINT "EINE{SPACE}VERAENDERUNG{SPACE}DER{SP
ACE}SYS{SPACE}ADRESSE                                           <225>
1960 PRINT "REICHT{SPACE}NICHT!                                 <207>
1970 PRINT                                                      <82>
1980 PRINT "DOUBLESCREEN{SPACE}BENUTZT{SPACE}DEN
{SPACE}FAC{SPACE}ALS                                            <173>
1990 PRINT "ZWISCHENSPEICHER.                                   <1>
2000 PRINT                                                      <112>
2010 PRINT "ES{SPACE}IST{SPACE}NUR{SPACE}LAUFFAE
HIG,{SPACE}WENN{SPACE}DIE                                       <186>
2020 PRINT "ROM-BANK{SPACE}EINGESCHALTET{SPACE}I
ST.                                                             <246>
2030 PRINT                                                      <142>
2040 PRINT "DIE{SPACE}DEMO-PROGRAMME{SPACE}ZEIGE
N{SPACE}NUR{SPACE}EINIGE                                        <208>
2050 PRINT "EFFEKTE.                                            <176>
2060 PRINT "SINNVOLLERE{SPACE}ANWENDUNGEN{SPACE}
INNERHALB{SPACE}VON                                             <218>
2070 PRINT "DATENVERARBEITUNGSPROGRAMMEN{SPACE}E
TC.                                                             <169>
2080 PRINT "BIETEN{SPACE}SICH{SPACE}NATUERLICH{S
PACE}AN!";                                                      <242>
2090 GOSUB 2130                                                 <114>
2100 PRINT CHR$(19)CHR$(19)                                     <25>
2110 RETURN                                                     <212>
2120 :                                                          <138>
2130 GETKEY A$                                                  <64>
2140 IF A$<>"{SPACE}" THEN 2130                                 <156>
2150 RETURN                                                     <252>
  ENDE DES LISTINGS
```

Big and Small

2 x Laufschrift für Ihre Programme

1. Small

Damit die Schriftzeilen in Ihren Programmen endlich Laufen lernen, hat uns Karsten Aprath das folgende kleine Programm geschrieben. Das Besondere an diesem Programm, ist die Tatsache, daß es vollständig mit unserem **Genesis-Super-Assembler** entwickelt wurde.

Es läßt sich in jedes Programm einfügen, sofern dieses nicht den Speicherbereich **16120 - 16383 (3EF8 - 3FFF)** überschreibt. Weiterhin wurde alles in diesem Bereich abgelegt und damit die Bereiche 1630 - 1771, usw. verschont.

Mit **SYS 16120** wird das Programm aufgerufen. Man befindet sich nun in einer **INPUT-Routine**. Alle Editierfunktionen verhalten sich, wie Sie es bei einem Inputbefehl gewöhnt sind.

Nun können Sie den Text eingeben, der im Laufschriftmodus angezeigt werden soll. Obwohl man mehr al 40 Zeichen eingeben kann, liest das Programm nur die ersten 40 Zeichen und legt sie in 16303 - 16342 ab. Nach dem RETURN meldet sich der Cursor zurück.

Die Laufschrift selbst wird mit **SYS 16182,x** gestartet. Das **x** (0 - 255) steht für die Geschwindigkeit der Laufschrift und wird in SKpeicherstelle **16261 (3F85)** geschrieben. Durch direkte Poken in diese Speicherstelle läßt sich die Geschwindigkeit jederzeit während des Betriebes ändern. Das Ausschalten der Laufschrift ist jederzeit mit **SYS 16169** möglich. Die Farbe der Zeichen läßt sich mit

POKE 16247,x

ändern.Für "x" steht hier 0 - 127 (normal) und 128 - 255 (blinkend).

Die Sterne (als Leerzeichen) kann man mit

POKE 16141,x

ändern. Für "x" setzen sie bitte den entsprechenden Wert aus der Poketabelle des Handbuches ein. Es ist danach allerdings en absoluter Neustart nötig, um die Zeichen in den vorgesehenen Speicherbereich zu schreiben.

***

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

10 REM==================================          <122>
20 REM=                                  =          <212>
30 REM=          LAUFSCHRIFT             =          <122>
40 REM=       BY KARSTEN APRATH          =          <231>
50 REM=    SCHOPENHAUERSTRASSE 25        =          <98>
60 REM=                                  =          <252>
70 REM==================================          <182>
80 POKE55,247:POKE56,62:CLR                          <134>
90 FORT=0TO139:READA$:A=DEC(A$):B=B+A:POKE16120+
T,A:NEXT                                             <132>
100 IFB<>16492THENPRINT"{CLEAR}FEHLER{SPACE}IN{S
PACE}DATA,S{SPACE}310-480":END                       <107>
110 FORT=16263TO16383:POKET,42:NEXT                  <247>
120 A=0:B=0:FORT=0TO39:READA$:A=DEC(A$):B=B+A:PO
KE16303+T,A:NEXT                                     <130>
130 IFB<>2631THENPRINT"{CLEAR}FEHLER{SPACE}IN{SP
ACE}DATA,S{SPACE}520-560":END                        <72>
140 SYS16182,10:PRINT"{CLEAR DOWN2 RIGHT SS}TART
{SPACE SS}ATZEINGABE{SPACE}DURCH{SPACE SS SY SS}
16120"                                               <101>
150 PRINT"{DOWN RIGHT SS}TART{SPACE SL}AUFSCHRIF
T{SPACE}DURCH{SPACE SS SY SS}16182,X"                <145>
160 PRINT"{DOWN RIGHT}X{SPACE}STEHT{SPACE}FUER{S
PACE}DIE{SPACE SG}ESCHWINDIGKEIT"                    <184>
170 PRINT"{DOWN RIGHT SA}USSCHALTEN{SPACE}DURCH{
SPACE SS SY SS}16169"                                <66>
180 PRINT"{DOWN RIGHT SS}CHRIFTFARBE{SPACE}DURCH
{SPACE SP}OKE16247,X{SPACE}AENDERN"                  <154>
190 PRINT"{DOWN RIGHT}X=0-127{SPACE SN}ORMAL{SPA
CE3}X=128-255{SPACE SB}LINKEND                       <153>
200 PRINT"{DOWN RIGHT SG}ESCHWINDIGKEIT{SPACE}DU
RCH{SPACE SP}OKE16261,1-255"                         <132>
210 PRINT"{DOWN RIGHT SD}URCH{SPACE SP}OKE16141,
X{SPACE}KOENNEN{SPACE}FUER{SPACE}DIE"                <168>
220 PRINT"{DOWN RIGHT SS}TERNE{SPACE}ANDERE{SPAC
E SZ}EICHEN{SPACE}GEPOKT{SPACE}WERDEN."              <190>
230 PRINT"{DOWN RIGHT SD}AS{SPACE SP}ROGRAMM{SPA
CE}MUSS{SPACE}DANN{SPACE}MIT{SPACE SS SY SS}1612
0"                                                   <135>
240 PRINT"{DOWN RIGHT}NEU{SPACE}GESTARTET{SPACE}
WERDEN"                                              <182>
250 DO:FORT=0TO255:POKE16247,T                       <237>
260 FORA=1TO200:GETA$:IFA$=""THENNEXTA:ELSESYS16
169:POKE16247,0:SCNCLR:END                           <132>
270 NEXTT:LOOP                                       <151>
280 REM==================================          <137>
290 REM      DATAS   MC PROGRAMM                     <127>
300 REM==================================          <157>
310 DATA20,29,3F,A9,8E,20,D2,FF                      <56>
320 DATAA9,08,20,D2,FF,A9,00,8D                      <178>
330 DATA84,3F,A2,78,A9,2A,9D,87                      <233>
340 DATA3F,CA,D0,FA,8D,87,3F,A2                      <74>
350 DATA00,20,CF,FF,C9,0D,F0,08                      <89>
360 DATA9D,AF,3F,E8,E0,28,D0,F1                      <86>
370 DATA60,78,A9,CE,A2,0E,8D,15                      <248>
380 DATA03,8E,14,03,58,60,20,D8                      <123>
390 DATA9D,8E,85,3F,8E,86,3F,A9                      <64>
400 DATA0E,20,D2,FF,78,A9,51,A2                      <122>
410 DATA3F,8D,14,03,8E,15,03,58                      <130>
420 DATA60,CE,86,3F,D0,15,AD,85                      <255>
430 DATA3F,8D,86,3F,EE,84,3F,AE                      <23>
440 DATA84,3F,E0,51,D0,08,A9,00                      <51>
450 DATA8D,84,3F,4C,0E,CE,A0,00                      <176>
460 DATABD,87,3F,99,00,0C,A9,00                      <186>
470 DATA99,00,08,E8,C8,C0,28,D0                      <27>
480 DATAEF,4C,0E,CE                                  <252>
490 REM==================================          <4>
500 REM      DATAS FUER DEMO                         <236>
510 REM==================================          <24>
520 DATA20,4C,41,55,46,53,43,48                      <31>
530 DATA52,49,46,54,44,45,4D,4F                      <196>
540 DATA20,28,43,29,20,42,59,20                      <19>
550 DATA20,4B,41,52,53,54,45,4E                      <104>
560 DATA20,41,50,52,41,54,48,20                      <248>
  ENDE DES LISTINGS
```

Big

8mal Vergrößerung auf dem C16

Dieses Programm vergrößert einen beliebigen Text 8mal und läßt diesen vom rechten in den linken Bildschirmrand scrollen. Zuerst muß der erste Teil geladen und gestartet werden, dann der zweite Teil. Der zweite Teil ist ein Editor, mit dem man einen beliebigen Text eingeben kann. Dieser Text wird dann, wenn man das Ganze mit SYS 14580 startet, 8mal vergrößert ausgegeben.

Diese und einige andere Informationen finden sie auch in den REM-Zeilen des 1. Programmes.

```
C-16 LISTING + CHECKSUMMEN (OC V1.0)

10 REM ******************************          <121>
20 REM *** MCODEMAKER FUER DAS PRO- ***          <59>
30 REM *** GRAMM 8 * VERGROESSERUNG ***          <69>
40 REM *** (C) BY TRONIC- VERLAG    ***          <227>
50 REM *** WRITTEN AND CREATED BY   ***          <148>
60 REM *** H A R T M U T   O T T    ***          <36>
70 REM *** T I G E R   -   S O F T  ***          <177>
80 REM ***                          ***          <77>
90 REM *** DIESES PROGRAMM VERGROE- ***          <156>
100 REM ** SERT DIE SCHRIFT 8 MAL   ***          <98>
110 REM ** UND LAESST DIESE VOM     ***          <140>
120 REM ** RECHTEN IN DEN LINKEN    ***          <118>
130 REM ** BILDSCHIRMRAND SCROLLEN. ***          <234>
140 REM ** DER TEXT DES SCHRIFTZUGES***          <176>
150 REM ** KANN IM NACHFOLGENDEN    ***          <11>
160 REM ** EDITOR ERZEUGT WERDEN .  ***          <153>
170 REM ** TIPPEN SIE NUN DIESES    ***          <42>
180 REM ** PROGRAMM SELBSTVERSTAEND-***          <135>
190 REM ** LICH OHNE REM - ZEILEN   ***          <146>
200 REM ** AB UND SPEICHERN SIE ES  ***          <21>
210 REM ** IM MONITOR MIT :         ***          <164>
220 REM ** S "B*VERGR.",1,3800,3988 ***          <180>
230 REM ** AB . LADEN KOENNEN SIE ES***          <60>
240 REM ** WIEDER !!AUSSERHALB!! DES***          <54>
250 REM ** MONITORS MIT LOAD "",1,1 ***          <215>
260 REM ** (NACH DEM LOADEN NEW EIN-***          <131>
270 REM ** GEBEN). DANN TIPPEN SIE  ***          <108>
280 REM ** DEN EDITOR AB UND SPEI-  ***          <237>
290 REM ** CHERN DIESEN EBENFALLS.  ***          <249>
300 REM ** NUN KOENNEN SIE DEN EDI- ***          <173>
310 REM ** TOR MIT RUN STARTEN UND  ***          <225>
320 REM ** IHREN GEWUENSCHTEN TEXT  ***          <118>
330 REM ** EINGEBEN (NOCH UEBER 1KB ***          <127>
340 REM ** SPEICHERPLATZ). DEN TEXT ***          <177>
350 REM ** UND DAS MCODEPROGRAMM    ***          <200>
360 REM ** KOENNEN SIE ZUSAMMEN IM  ***          <122>
370 REM ** MONITOR MIT :            ***          <12>
380 REM ** S "B*VERGR.",1,3800,3E80 ***          <165>
390 REM ** ABSAVEN UND JE NACH BE-  ***          <176>
400 REM ** DARF WIEDER WIE OBEN LA- ***          <8>
410 REM ** DEN. GEFALLEN IHNEN DIE  ***          <209>
420 REM ** FARBEN DES PROGRAMMS     ***          <3>
430 REM ** NICHT SO KOENNEN SIE DIE ***          <162>
440 REM ** SE DURCH FOLGENDE POKES  ***          <22>
450 REM ** AENDERN :                ***          <43>
460 REM ** POKE14617,ZEICHENFARBE   ***          <43>
470 REM ** POKE14646,HINTERGRUNDF.  ***          <251>
480 REM ** POKE14651,RAHMENFARBE    ***          <144>
490 REM ** POKE14667,SCHNELLIGKEIT  ***          <166>
500 REM ** DES SCROLLINGS . WERTE : ***          <53>
510 REM ** (1=SEHR SCHNELL - 255=   ***          <111>
520 REM ** SEHR LANGSAM)            ***          <236>
530 REM ** UND NUN VIEL SPASS       ***          <129>
540 REM ******************************          <35>
60000 DATA00,00,00,00,00,A9,0D,85, 315           <121>
60001 DATAFE,A9,68,85,FD,A0,01,B1, 1251          <165>
60002 DATAFD,88,91,FD,C8,C8,C0,28, 1419          <184>
60003 DATAD0,F5,A5,FD,18,69,28,90, 1184          <97>
60004 DATA02,E6,FE,85,FD,C9,A8,D0, 1449          <253>
60005 DATAE4,A5,FE,C9,0E,D0,DE,AD, 1465          <179>
60006 DATA00,38,85,FD,AD,01,38,85, 805           <29>
60007 DATAFE,A0,00,8C,04,38,B1,FD, 1044          <180>
```

Wettbewerb 1/87

Keine Preisfrage, kein Rätsel, kein Puzzle wollen wir diesmal mit Ihnen veranstalten. Sondern wir wollen von Ihnen einige Fragen über Ihren Computer beantwortet wissen. Sie helfen uns damit bei der Planung für die zukünftigen Hefte und können dann vielleicht »Compute mit«-Hefte in der Hand halten, die noch mehr Ihr Gefallen finden.
Außerdem honorieren wir 50 Einsender, die wir auslosen werden, mit tollen Preisen!

1. Preis
Eine Plus 4/Floppy 1551-Kombination

2. bis 10. Preis
Ein Joystick für C 16/116/plus 4 (gestiftet von T.S. Datensysteme)

11. bis 20. Preis
Programmpaket aus dem aktuellen Softwaremarkt

21. bis 50. Preis
Programmpaket Ihrer Wahl aus unserem Softwareservice (Seite 34)

1. Ich besitze einen Computer vom Typ

_____

2. Ich besitze diesen Computer seit (Monat/Jahr)

_____

3. Ich lese die monatliche Zeitschrift »Compute mit«
   - ☐ im Abo ☐ gelegentlich
   - ☐ regelmäßig ☐ nie
   - ☐ die Monatszeitschrift »Compute mit« kenne ich noch nicht.

4. »Compute mit« und »Compute mit-Sonderheft«...
   - ☐ erwerbe ich selbst ☐ lese bei Freunden mit

5. Ich interessiere mich hauptsächlich für
   - ☐ Spiele ☐ Anwenderprogramme
   - ☐ Tips & Tricks ☐ Kurse

6. Längere Programme aus den Heften
   - ☐ tippe ich selbst ab
   - ☐ lasse abtippen
   - ☐ besorge mir von Freunden
   - ☐ bestelle ich beim Software-Service

7. Sollte ich einmal einen neuen Computer erwerben, wird es ein

_____

sein!

Wir versichern, daß Ihre Daten nicht mit Namen erfaßt werden. Auch eine Weitergabe an Dritte ist ausgeschlossen.

Ausschneiden und abschicken an:

Tronic-Verlag
Postfach 870
3440 Eschwege

Absender:

Name: _____

Straße: _____

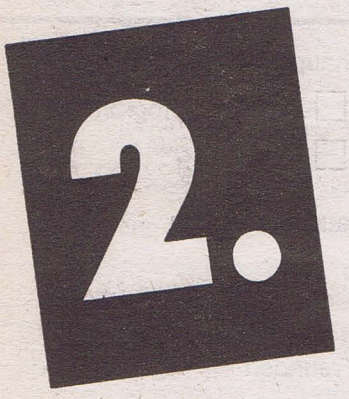
Stadt: _____

software-service

Alle Programme dieses Heftes können Sie als Programmpaket auf Diskette oder Kassette erwerben.

Sonderheft 1/87

Bestell-Nr. CSOK-1/2 Kassette 35,– DM
Bestell-Nr. CSOD-1/2 Diskette 35,– DM



Auch die Programmpakete für die Sonderhefte 1, 2 + 3 sind noch zu haben!

Sonderheft 1/86: (Heft 6,50 DM)
Bestell-Nr. CSOK-1 Kassette 30,– DM
Bestell-Nr. CSOD-1 Diskette 30,– DM

Sonderheft 2/86: (Heft 6,50 DM)
Bestell-Nr. CSOK-2 Kassette 30,– DM
Bestell-Nr. CSOD-2 Diskette 30,– DM

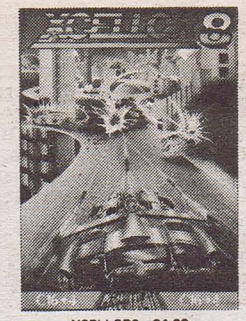
Sonderheft 3/86: (Heft 6,50 DM)
Bestell-Nr. CSOK-3 Kassette 30,– DM
Bestell-Nr. CSOD-3 Diskette 30,– DM

Bestellungen richten Sie bitte an:
Tronic-Verlag · Postfach 870 · 3440 Eschwege
oder telefonisch: (0 56 51) 3 00 11
Der Versand erfolgt per Nachnahme
oder Vorkasse (Scheck, Bar)

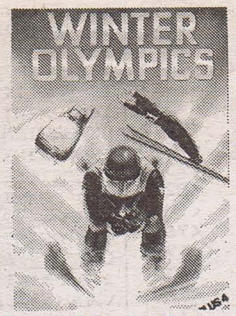
Eine Bitte: Der Softwareversand ist ein Leserservice des Tronic-Verlags. Alle Anleitungen sind daher dem entsprechenden Heft zu entnehmen. Wollen Sie ältere Software, bestellen Sie bitte das Heft mit. Sollte es ausverkauft sein, kopiert unser Versand gegen einen Unkostenbeitrag in Höhe des Heftpreises alle Anleitungen für Sie. Vermerken Sie bitte auf Ihrer Bestellung »+ Heft«!

T.S. Datensysteme-Vertriebsgesellschaft mbH

Yie ar Kung Fu **29,90**



XCELLOR8 **24,90**



Winter Olympics **27,90**

Trizons **14,90**

Mounty on the run **23,90**

Berks I, II, III **25,90**

Space Pilot **19,90**

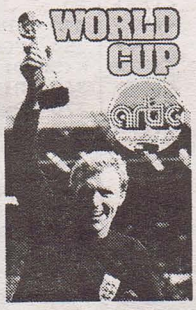
Video Poker **19,90**

Sport 4 **24,90**

Ghosts'n Goblins **29,90**

Kikstart **9,90**

A.C.E. **35,90**

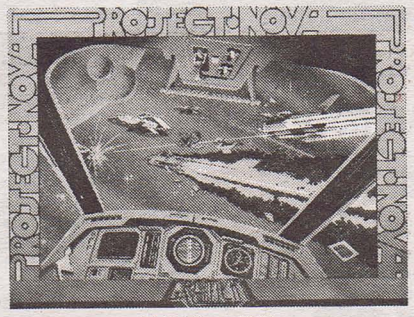


World Cup **19,90**

Jump Jet **39,90**

Thai Boxing **19,90**

Winter Events **24,90**



Project Nova **24,90**

Frank Bruno's Boxing **19,90**

Wimbledon **29,90**

Star Events **25,90**

Air Wolf	29,90
Atlantis	14,90
Bandits at zero	14,90
Beach-Head	29,90
Big Mac	9,90
Bomb Jack	29,90
Bongo	19,90
Bridgehead	29,90
Cave Fighter	9,90
Classics III, 4-Games	27,90
Classics II, Cass. mit 4 Prog.	
Commando	25,90
Computer Hits 3, 10-Games	27,90
Cruncher	19,90
Death Race 16	14,90
European Games	24,90
Fingers Malone	9,90
Five Star, 5-Games	
Flight Path 737	19,90
Formula 1	9,90
Galaxians	19,90
Ghost Town	19,90
Grand Master Chess	29,90
Greatest Hits, Cass. mit 4 Prog.	
Gullwing Falcon	25,90
Gunslinger	19,90
Hektik	9,90
Home Office	35,90
Hustler	19,90
International Karate	29,90
Invasion 2000 AD	28,90
Jet Set Willy	28,90
Jetprix	24,90
King Size Daten Bank	24,90
King Size Turbotext	24,90
Killapede	9,90
King Size 50 Games	26,90
Knock Out	14,90
Kung Fu Kid	19,90
Lawn Tennis	24,90
Leapin Louie	19,90
Legionnaire	25,90
Manic Miner	22,90
Mr. Puniverse	9,90
One Man Droid	9,90
Paintbox	19,90
Petch	14,90
Pogo Pete	19,90
Powerball	9,90
Reach for the Sky	19,90
Robin to the Rescue	9,90
Skyhawk	9,90
Slipery Sid	14,90
Snooker	29,90
Speed King	9,90
Street Olympics	9,90
Tazz	19,90
The Return of Rockman	9,90
The Wizard & the Princess	19,90
Tomp of Tarrabash	19,90
Trailblazer	29,90
World Cup Carnival	29,90
Winie Witch's	
Superbroom	14,90
World Series Baseball	29,90
Competition Pro 3000	
Joystick	19,90
Competition Pro 5000	
Joystick	29,90
Medalist mit Autofeuer	15,90
Quickshot I Joystick	9,90
Speed King Joystick	29,90

Denisstraße 45, 8500 Nürnberg 80, Tel. 0911/288286

```
60008 DATA0A,2E,04,38,0A,2E,04,38, 232      <6>
60009 DATA0A,2E,04,38,EA,85,FD,AD, 909      <6>
60010 DATA04,38,18,69,D0,85,FE,A5, 949      <252>
60011 DATAFE,C9,D4,90,05,A2,01,4C, 1055     <143>
60012 DATA7E,39,A2,00,8E,02,38,A2, 707      <9>
60013 DATA00,A0,00,AD,02,38,C9,01, 593      <245>
60014 DATAF0,17,B1,FD,2D,03,38,F0, 1037     <214>
60015 DATA08,A9,66,9D,8F,0D,4C,9D, 825      <154>
60016 DATA38,A9,20,9D,8F,0D,4C,9D, 803      <221>
60017 DATA38,B1,FD,2D,03,38,F0,08, 838      <79>
60018 DATAA9,20,9D,8F,0D,4C,9D,38, 803      <233>
60019 DATAA9,66,9D,8F,0D,8A,18,69, 851      <82>
60020 DATA28,AA,C8,C0,07,D0,C4,B1, 1190     <109>
60021 DATAFD,2D,03,38,F0,15,AD,02, 793      <76>
60022 DATA38,F0,08,A9,20,8D,A7,0E, 827      <179>
60023 DATA4C,D5,38,A9,66,8D,A7,0E, 938      <230>
60024 DATA4C,D5,38,AD,02,38,F0,08, 824      <162>
60025 DATAA9,66,8D,A7,0E,4C,D5,38, 938      <118>
60026 DATAA9,20,8D,A7,0E,4E,03,38, 660      <204>
60027 DATAAD,03,38,F0,01,60,EE,00, 807      <48>
60028 DATA38,D0,03,EE,01,38,AD,01, 736      <26>
60029 DATA38,C9,3F,D0,02,A9,3A,4C, 833      <146>
60030 DATA57,39,EA,EA,A9,3A,8D,01, 981      <119>
60031 DATA38,A9,00,8D,00,38,A9,80, 719      <86>
60032 DATA8D,03,38,A9,0C,85,FE,A9, 937      <192>
60033 DATA00,85,FD,A0,00,A9,20,91, 892      <64>
60034 DATAFD,A5,FE,38,E9,04,85,FE, 1352     <105>
60035 DATAA9,77,91,FD,A5,FE,18,69, 1234     <156>
60036 DATA04,85,FE,E6,FD,D0,02,E6, 1314     <187>
60037 DATAFE,A5,FD,C9,E8,D0,DC,A5, 1698     <177>
60038 DATAFE,C9,0F,D0,D6,A9,90,8D, 1346     <47>
60039 DATA15,FF,A9,F3,8D,19,FF,20, 1141     <59>
60040 DATA05,38,A2,00,A0,00,C8,D0, 791      <160>
60041 DATAFD,E8,E0,05,D0,F6,AD,00, 1341     <85>
60042 DATA38,85,FD,4C,60,39,EA,8D, 1046     <97>
60043 DATA01,38,A9,80,8D,03,38,60, 650      <238>
60044 DATAAD,01,38,85,FE,A0,00,B1, 954      <152>
60045 DATAFD,C9,60,D0,0A,A9,3A,8D, 1136     <197>
60046 DATA01,38,A9,00,8D,00,38,A5, 588      <239>
60047 DATAC6,C9,40,F0,C2,60,A5,FE, 1412     <132>
60048 DATA38,E9,04,85,FE,4C,64,38, 912      <169>
60049 DATA00,00,00,00,00,00,00,00, 0        <206>
60050 FORT= 14336 TO 14728 STEP8:P=0        <185>
60051 FORI=0TO7:READA$:A=DEC(A$):P=P+A:POKET+I,A
:NEXTI                                      <237>
60052 READR:IFP<>RTHENPRINT"PRUEFSUMMENFEHLER{SP
ACE}IN{SPACE}ZEILE"PEEK(63)+PEEK(64)*256:END <86>
60053 NEXT:PRINT"DATAS{SPACE}SIND{SPACE}RICHTIG{
SPACE}!!!"                                  <123>
  ENDE DES LISTINGS
```

Editor

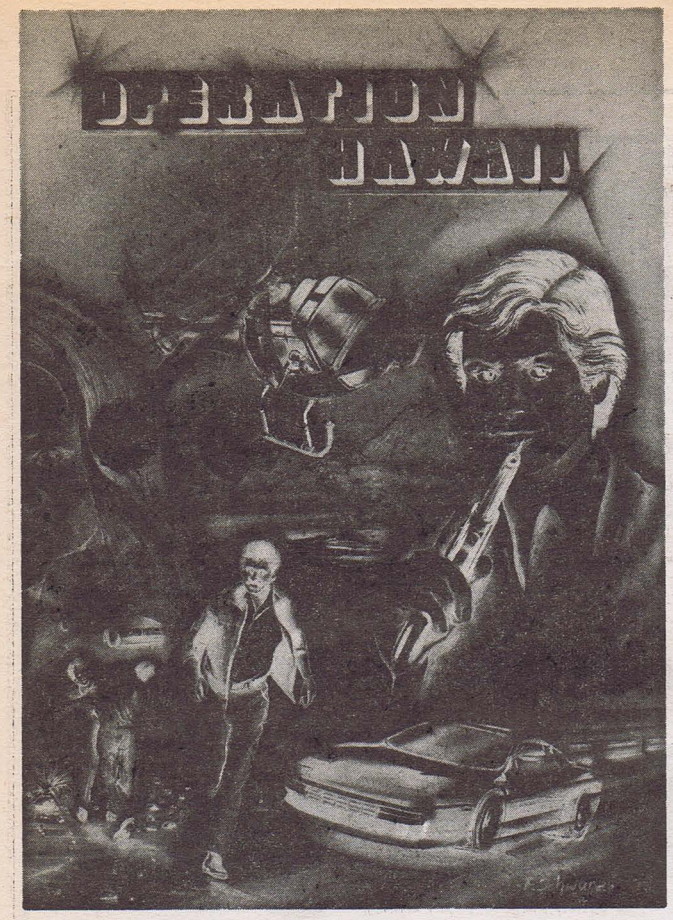
```
  C-16 LISTING + CHECKSUMMEN (OC V1.0)

10 REM ******************************      <121>
20 REM ***SCHRIFT-VERGROESSERER  ******     <123>
30 REM ***    II. T E I L        ******     <253>
40 REM ***                       ******     <147>
50 REM ***    E D I T O R        ******     <16>
60 REM ***                       ******     <167>
70 REM ***  BY HARTMUT OTT       ******     <24>
80 REM ***  T I G E R - SOFT     ******     <21>
90 REM ******************************      <201>
100 COLOR0,9,4:COLOR4,1,1                   <41>
110 POKE65299,208:POKE1351,128              <103>
120 PRINT"{CLEAR BLACK RIGHT3 DOWN2}DIES{SPACE}I
ST{SPACE}DER{SPACE}EDITOR{SPACE}ZUM{SPACE}VORHER
-"                                          <203>
130 PRINT"{RIGHT3 DOWN}GEHENDEN{SPACE}M.PROGRAMM
{SPACE}.{SPACE}TIPPEN{SPACE}SIE"            <120>
140 PRINT"{RIGHT3 DOWN}NUN{SPACE}IHREN{SPACE}GEW
UENSCHTEN{SPACE}TEXT{SPACE}EIN."            <253>
150 PRINT"{RIGHT3 DOWN}ACHTUNG:{SPACE}BENUETZEN{
SPACE}SIE{SPACE}NICHT{SPACE}DIE"            <2>
160 PRINT"{RIGHT3 DOWN}TASTEN{SPACE}ESC,CLR{SPAC
E}HOME{SPACE}RETURN"                        <96>
170 PRINT"{RIGHT3 DOWN}F1-F7{SPACE}UND{SPACE}CTR
L{SPACE}AUSSER{SPACE}IN{SPACE}VER-"         <128>
180 PRINT"{RIGHT3 DOWN}BINDUNG{SPACE}MIT{SPACE}R
VSON{SPACE}UND{SPACE}RVSOFF"                <241>
190 PRINT"{RIGHT6 DOWN2}T{SPACE}A{SPACE}S{SPACE}
T{SPACE}E{SPACE2}DRUECKEN"                  <180>
200 POKE239,0                               <60>
210 IFPEEK(198)=64THEN210                   <201>
220 POKE239,0                               <80>
230 SP=14848:RE=0                           <48>
240 SCNCLR                                  <216>
250 PRINT"{HOME WHITE RIGHT6 DOWN3}SPEICHERSTELL
E{SPACE}:{SPACE}";SP                        <100>
260 PRINT"{HOME RIGHT6 DOWN4}HOECHSTE{SPACE}SPEI
CHERSTELLE{SPACE}16000"                     <28>
270 PRINT"{HOME RIGHT6 DOWN5}MIT{SPACE}SHIFT+SPA
CE{SPACE}ENDE"                              <123>
280 PRINT"{HOME RIGHT18 RVSON SPACE RVSOFF}";  <24>
290 GETA$:IFA$=""THEN290                    <243>
300 IFA$=CHR$(20)THEN390                    <99>
310 IFA$="{RVSON}"THENRE=1:GOTO290          <88>
320 IFA$="{RVSOFF}"THENRE=0:GOTO290         <86>
330 PRINT"{HOME RIGHT18}"A$:IFPEEK(3090)=96THEN4
20                                          <54>
340 IFRE=0THENPOKESP,PEEK(3090)             <179>
350 IFRE=1THENPOKESP,PEEK(3090)+128:POKE3090,PEE
K(3090)+128                                 <164>
360 PRINT"{HOME RIGHT}";CHR$(20)            <21>
370 SP=SP+1:IFSP=16000THEN420               <71>
380 POKE51,176:POKE52,63:GOTO250            <251>
390 IFSP=14848THEN250:ELSEPOKE3090,32:PRINT"{HOM
E INST}":SP=SP-1                            <104>
400 IFSP-17>14848THENPOKE3072,PEEK(SP-18):ELSEPO
KE3072,32                                   <24>
410 GOTO380                                 <242>
420 POKESP,96                               <227>
430 PRINT"{CLEAR RIGHT4 DOWN2}NUN{SPACE}KOENNEN{
SPACE}SIE{SPACE}DEN{SPACE}TEXT"             <150>
440 PRINT"{RIGHT4}MIT{SPACE}DEM{SPACE}MCODEPROGR
AMM{SPACE}(SOFERN{SPACE}DIES"               <157>
450 PRINT"{RIGHT4}IM{SPACE}SPEICHER{SPACE}IST){S
HIFTSPACE}ABSAVEN."                         <103>
460 PRINT"{RIGHT4}SIE{SPACE}KOENNEN{SPACE}DIESES
{SPACE}ABGESPEICHERTE"                      <197>
470 PRINT"{RIGHT4}PROGRAMM{SPACE}DANN{SPACE}JEDE
RZEIT{SPACE}WIEDER"                         <171>
480 PRINT"{RIGHT4}LADEN{SPACE}UND{SPACE}MIT{SPAC
E}SYS{SPACE}14580{SPACE}STARTEN."           <140>
490 REM ******************************      <240>
500 REM ****** ENDE DES LISTINGS ******     <179>
510 REM ******************************      <4>
  ENDE DES LISTINGS
```

monatlich

Compute mit

COMMODORE & SCHNEIDER



OPERATION HAWAII

Schlüpfen Sie in die Rolle eines Privatdetektives und übernehmen Sie einen atemberaubenden Fall. Wohin verschwand der berühmte Völkerkundler C. D. Allen?

Warum hinterließ seine letzte Expedition keine Spuren und was ist über diese Expedition bekannt?

Nur Sie können das Rätsel lösen! Dabei werden Sie immer wieder in brenzlige, teilweise auch humorvolle Situtationen geraten, die Ihr Geschick auf die Probe stellen. Zwei Dinge sind wichtig: Niemals die Nerven verlieren und cool bleiben!

Für Plus 4 oder C-16 mit 64 KB-RAM
Kassette

Nur 29,–

MICRO=HÄNDLER

64 K-RAM
Speichererweiterung

- 60671 BASIC BYTES FREE
- Einstecken und fertig! Keine Umbauten, da Steckmodul.

COMMODORE C-16

C-16 Speichererweiterung 64 KB-RAM

Einfach einstecken und fertig!
Keine Umbauten oder Lötarbeiten notwendig, da Steckmodul.
Volle 60671 Basic Bytes frei.

Zusammen mit dem Adventure
OPERATION HAWAII

Nur 139,–

Neu! HOLLYWOOD POKER gibt es jetzt auch für den Plus 4! Werfen Sie hierzu einen Blick in die ASM! Versand erfolgt grundsätzlich nur per Nachnahme zuzüglich 5,– DM Portospesen!

Auch im führenden Fachhandel sowie in allen Kaufhäusern!

Softwareautoren gesucht!
Händleranfragen erwünscht!

Im Exklusivvertrieb von:

MICRO=HÄNDLER

Mitvertrieb:

RUSH WARE
Online with the trend.

Ein Produkt von

Eckhardt & Gehrmann GbR
Heinrichstraße 25
3000 Hannover 1
Tel. 0511/344771

Hot-line:

Bei Fragen zu den Listings steht Ihnen unsere Programmierabteilung für telefonische Anfragen zur Verfügung. Bitte wählen Sie nur diese Nummer und rufen Sie nur zu den angegebenen Zeiten an: Montag-Freitag, 14-16 Uhr!

0 56 51/3 00 13

Einnahmen – Ausgaben – Kontoführungen für C 16/116/plus 4

Kundenrechnungen

Dieses Verarbeitungsprogramm kann für die monatliche Abrechnung benutzt werden. Im Programm kann man diverse Einnahmen und Ausgaben festhalten. Die verschiedenen Programmteile, die angesprochen werden können, werden in den folgenden Abschnitten erklärt.

Alte Werte laden

In diesem Programmteil wird ein vorher gesicherter SEQ.-FILE nachgeladen, der User wird aufgefordert, das Band richtig zu spulen und "PLAY" zu drücken.

Werte Berichtigen / Löschen

Dieser Programmteil ermöglicht, versehentlich eingetippte Fehler zu beseitigen. Man hat die Auswahl, einen Wert zu löschen oder zu berichtigen, der auf der Einnahmen- (E) oder Ausgaben- (A) Seite steht. Dazu muß man sich die jeweilige Zahl merken, die beim Bildschirmausdruck gezeigt wird, und sie eingeben.

Erfassung (E/A)

Bei diesem Programmteil werden die Einnahmen und Ausgaben erfaßt. Es stehen vier Möglichkeiten zur Wahl:

E: Erfassung der Einnahmen
A: Erfassung der Ausgaben
Q: Zurück zum Menü
(Return):
Sprung zur Ausdruck- Abfrage

E = Erfassung Einnahmen

Hier wird zuerst die Belegnummer abgefragt, die nur numerisch sein darf. Danach wird der Betrag der Einnahme abgefragt. Durch (Return) Rücksprung zu Abfrage (E/A/Q).

A = Erfassung Ausgaben

Das Gleiche wie (E), nur das die Werte der Ausgaben abgefragt werden.

(RETURN) = Sprung Ausdruck-Abfrage

Hier wird abgefragt, ob ein Ausdruck auf dem Drucker (D) oder Bildschirm (B) erfolgen soll. Wichtig! Vor dem Ausdruck auf dem Drucker muß auf dem Bildschirm ausgedruckt werden, weil sonst die Variablen nicht die richtigen Werte besitzen.

Werte sichern

Hier können die Einnahmen und Ausgaben gesichert werden.

Werte ausdrucken

Bei dieser Abfrage wird direkt in die Ausdruck-Abfrage gesprungen.

Programm beenden

Das Programm wird einfach beendet.

C-16 LISTING + CHECKSUMMEN (DC V1.0)

```
10 DIMZ(100),S(100),W(100),D(100),U(100),B(100),
L(100),LI(61)                                          <5>
20 DIMBE1(100),BA1(100),H1(100),H2(100)                <161>
30 PRINT"{CLEAR}":INPUT"DATUM{SPACE}[TT.MM.JJ]";
TAG$                                                   <226>
40 PRINT"{CLEAR}"                                      <70>
50 PRINT"{CLEAR UP2 L.BLU RVSON SPACE4 RVSOFF BL
ACK}EINNAHMEN{SPACE}-{SPACE}AUSGABEN{SPACE}BEREC
HNUNG{L.BLU RVSON SPACE5 RVSOFF BLACK}":R=0:P=0:
M=0:O=0                                                <139>
60 PRINT"{L.BLU RVSON SPACE40 RVSOFF BLACK}"           <116>
70 PRINT"{HOME DOWN6 RIGHT2}[1]{SPACE}={SPACE}AL
TE{SPACE}WERTE{SPACE}LADEN"                            <108>
80 PRINT"{HOME DOWN7 RIGHT2}[2]{SPACE}={SPACE}WE
RTE{SPACE}BERICHTIGEN/LOESCHEN"                        <16>
90 PRINT"{HOME DOWN8 RIGHT2}[3]{SPACE}={SPACE}ER
FASSUNG{SPACE}[E/A]"                                   <52>
100 PRINT"{HOME DOWN9 RIGHT2}[4]{SPACE}={SPACE}W
ERTE{SPACE}ABSAVEN"                                    <90>
105 PRINT"{HOME DOWN10 RIGHT2}[5]{SPACE}={SPACE}
WERTE{SPACE}AUSDRUCKEN"                                <248>
110 PRINT"{HOME DOWN11 RIGHT2}[9]{SPACE}={SPACE}
PROGRAMM{SPACE}BEENDEN"                                <65>
120 GETKEYK$:IFK$="1"THEN900                           <191>
130 IFK$="2"THENGOTO260                                <42>
140 IFK$="3"THENGOTO190                                <81>
150 IFK$="4"THENGOTO1140                               <169>
```

C-16/116/+4 »News«

Super Joystick

Nicht ganz billig ist dieser Joystick, der vom Ing. Büro Stechmann vertrieben wird. Aber in allen Spielsituationen war er wirklich sein Geld wert. Berührungsfreie Schalter erlauben ein unheimlich sauberes Steuern und schnelles Reagieren. Immerhin 98,- DM muß man dafür hinblättern, allerdings sollten man bedenken, daß die Kontakt- und Mikroschalter anderer Joysticks der 30,-DM-Klasse nur begrenzte Lebensdauer aufweisen. Außerdem fanden wir, daß der Spielspaß mit diesem Joystick erheblich zunimmt. Die Federkraft des Knüppels läßt sich übrigens durch drehendesselben regulieren. Erhältlich für Atari ST, Amiga, C 64, MSX, C16-Version in Vorbereitung.
Info: Marathon-Joystick
Vertrieb: ifi Stechmann, PF 210, 2152 Horneburg

Speichererweiterung als Einbauservice

ifi Stechmann bietet den einbau einer 64k-Speichererweiterung für C16/116 ab sofort 99,-DM an. Als besonderes Bonbon wird eins der leistungsfähigsten 64K-Programme kostenlos mitgeliefert. Und zwar entweder ACE, MERCENARY oder SCRIPT plus.
Info: s.o.

Programm des Jahres:
WINTER OLYMPIADE

Das C-16/116 und Plus 4 Programm des Jahres heißt **WINTER OLYMPIADE**. Die englische Fachzeitschrift COMMODORE COMPUTING INTERNATIONAL zeichnete dieses Programm mit insgesamt 3 Oskars aus. Zum erstenmal gelang es einem deutschen Software-Haus diese begehrten Trophäen zu erkämpfen. Jeden Insider der Software-Szene dürfte dieser Erfolg ein wenig überrascht haben, war man doch bisher gewohnt, das die nationalstolzen Engländer mit der Bewertung ausländischer Software immer äußerst skeptisch umgingen. Nachdem die gesamte deutsche Fachpresse aber von diesem Programm und seiner hervorragenden Qualität überzeugt war, ist man auch in England letztendlich zu dem Ergebnis gekommen, daß dies wohl das beste Programm für den C-16/116 und Plus 4 ist, das im vergangenen Jahr vorgestellt wurde. So ist also der Titel **"Programm des Jahres"** eine besondere Auszeichnung. Doch der Lorbeeren nicht genug. Es gesellten sich gleich noch zwei weitere Oskars hinzu. Udo Gertz wurde von der Fachjurie zum **"Programmierer des Jahres"** gekürt. Er hat nach Meinung vieler mit diesem Programm neue Maßstäbe im Bereich der Programmierung des "kleinen Commodore" gesetzt und sollte dafür auch die ihm gebührende Anerkennung finden. Und der dritte und damit letzte Oskar für dieses Programm wurde für die **"Grafik des Jahres"** vergeben. Nach Meinung aller Juroren ist hier eine Grafik geschaffen worden, die alles bisher Dagewesene in den Schatten stellt. Wir möchten uns an dieser Stelle allen Gratulanten anschließen und einen Glückwunsch und zugleich ein Dankeschön an das Softwarehaus Kingsoft und alle Beteiligten weiterreichen. Das Dankeschön übermitteln wir im Namen aller Anwender, die natürlich hoffen, daß dieser Erfolg Ansporn dafür ist, damit auch in Zukunft ähnlich gute und erfolgreiche Programme produziert werden können. Die Redaktion empfiehlt an dieser Stelle allen Usern die noch nicht im Besitz der "Winter Olympiade" sind, sich noch schnell eine Version zu sichern.

Ladeprobleme nun endlich gelöst?

Programm: CSJ Turbo
System: C16/Plus 4
Preis: 29.90 DM
Hersteller: CSJ Computersoft Jonik, An der Tiefenriede 27, 3000 Hannover 1, Telefon 0511/886383

Jeder Besitzer einer Commodore Datasette kennt sicherlich die gefürchtete Fehlermeldung "LOAD ERROR". Diese Meldung des Computers weist darauf hin, das ein Ladefehler entanden ist und somit das Programm nicht geladen werden kann. Das Problem tritt besonders oft auf, wenn man versucht Programme zu laden welche von einem anderen Computer bzw. einer anderen Datasette abgespeichert wurden. In der Regel ist hier eine Fehlerhafte Einstellung des Tonkopfes sowie das relativ unsichere Ladeverfahren des C16 schuld. Obwohl Datasette eigendlich speziell für Commodore Rechner entwickelt wurden, bestehen diese meißt nur aus einer billigen Elektronik. Auch die Einstellung des Tonkopfes ist je nach Hersteller, es gibt verschiedene Produktionsstätten (z.B. Japan o. Taiwan), offensichtlich unterschiedlich. Um diesen die Datasetten genauer zu Justieren benötigt man normalerweise ein Ozzilloskop und eine entsprechende Justierkassette. Da diese Werkzeuge wohl nur den wenigsten Computer-Freunden zur Verfügung stehen, hat die Firma CSJ ein Programm herausgebracht, welche auch dem Laien die Einstellung des Tonkopfes erlauben soll. Das Programm nennt sich AZIMUTH-TAPE und wird zusammen mit einem TURBO-TAPE Verfahren auf einer Kassette geliefert. Nachdem das Programm AZIMUTH getartet wird erscheint eine Art Skale am unteren Bildschirmrand. Läßt man nun die Kassette weiterlaufen, so flackert der Bildschirmrand und ein entsprechender Messzeiger. Ein Signalton soll nun dafür sorgen, das man die Datasette richtig justieren kann. Um dies zu bewerkstelligen dreht man, während das Band noch läuft, vorsichtig den Tonkopf nach lings bzw. rechts. Ziel ist es zu erreichen das der Bildschirmrand möglicht ein ruhiges blau, und der Messzeiger möglichst garnichts, anzeigt. Wir haben diesen Einstellung vorgenommen und das Ergebnis mit einem Ozilloskop überprüft. Es ist uns tatsächlich gelungen fast die ideal Einstellung zu finden. Allerdings gehört dazu schon etwas Fingerspitzengefühl und auch etwas Glück, den das Signal, nach welchem das Programm die Datasette untersucht ist digitalisiert. Dies bedeutet das kleine Pegelunterschiede nicht festgetellt werden können, also nur eine Art Mittelwert eingestellt werden kann. Dies dürfte in der Praxis immer noch besser sein als überhaupt keine Einstellungsmöglichkeit.
Das zweite Problem des C16, die lansame Datenübertragung, wird durch das Programm CSJ TURBO ausgeglichen. Im Vergleich zum normalen Lade/Speicher Verfahren wird mittels TURBO bis zu einer 12 fachen Geschwindigkeit gearbeitet, was sogar eine Konkurrenz zum Floppy-Laufwerk bedeutet. Nachdem TURBO aktiviert ist arbeiten somit die Befehle LOAD und SAVE mit der 12 fachen Geschwindigkeit. Dateiverarbeitung mittels OPEN wird alldings von diesem Programm nicht untertützt. Interessant ist auch das Ladeverfahren, nachdem der Vorspann geladen wurde, wird Programmname, Programmlänge sowie Programmende in HEX und Dezimal angezeigt.
Der einzige Nachtel von CSJ TURBO ist allerdings das Programme, welche mit dem Programm abgespeichert wurden, auch nur mit diesem wieder eingeladen werden können.
Frank Brall

Positiv:
TURBO TAPE erreicht bis zu 12 fache Geschwindigkeitssteigerung
Einfache Tonkopfeinstellung ohne Messgerät
Günstiges Preis/Leistungsverhältnis.
Negativ:
Tonkopfeinstellung nicht ganz genau Möglich
Programme welche mit TURBO abgespeichert wurden, können auch nur mit TURBO wieder eingeladen werden

Genesis-Super-Assembler

Der große Erfolg unseres Assemblers für den C-16 hat uns bewogen, dieses nützliche und für jeden C-16-Programmierer unbedingt notwendige Utility ab sofort auch separat zu vertreiben. Der komplette Assembler wird inclusive Demoprogramm und ausführlicher Anleitung entweder auf Diskette oder Tape geliefert. Tatsächlich ist GENESIS momentan der leistungsstärkste Assembler für den C-16/116/plus 4 auf dem gesamten Softwaremarkt. Ein ganze Reihe von Features erleichtern dabei das Programmieren in Maschinensprache ungemein. Das besondere Bonbon für die Benutzer von nicht erweiterten Systemen C-16/116 ist die Möglichkeit, Quelltexte ,die in Bearbeitung sind, auszulagern (auf Diskette oder Tape) ,um so wertvollen Speicherplatz zu sparen.

Genesis-Super-Assembler kostet 15,– DM sowohl für Kassette wie auch Diskette und ist beim Tronic-Verlag, PF 870, 3440 Eswchwege (Tel.:05651/30011) zu beziehen.

Neues Hardware-Angebot zur Weihnachtszeit

Rechtzeitig zum Weihnachtsgeschäft finden Sie in vielen Kaufhäusern und in einigen größeren Verbrauchermärkten wieder ein Super-Angebot von Commodore-Rechner. Der Plus 4 wird einmal zusammen mit der Floppy-Disk 1551 zum Preis von ca. 500,– DM und einmal zusammen mit einer Datasette zum Preis von ca. 250,– DM angeboten. Im Hause Commodore rechnet man mit ähnlichen Absatzerfolgen wie bei den vorangegeangen Verkaufsaktionen des C-16/116. Nutzen Sie diese günstigen Verkaufsaktionen, aber achten Sie auch auf die mitgelieferte Hardware, damit Sie keine bösen Überraschungen zum Fest erleben, weil für Ihren Bedarf den falschen Computer gekauft haben.

Deutscher Zeichensatz
Einen deutschen Zeichensatz für den C-16/116/plus 4 kann man jetzt als Eprom erwerben. Einfach einstecken, schon kann man mit dem mitgelieferten Schalter zwischen ASCII und deutscher DIN-Norm wählen, genauso, wie es bei dem Commodore 128'er ist. Im Lieferumfang sind Tastenaufdrucke mit den Doppelbelegungen enthalten. Kingsoft

Komplette Textverarbeitung mit dem C 16

Programm: Text C16
System: C16
Preis: 29.90 DM
Hersteller: CSJ Computersoft Jonigk, An der Tiefenriede 27, 3000 Hannover, Tel.0511/886383

Nach dem es Anfangs noch relativ wenige brauchbare Programme für den kleinen C16 gab, häufen sich in den letzten Monaten die Angebote von preißwerter Software. Auch für den Anwender welcher den C16 als günstiges Textsystem einsetzen möchte, stehen mittlerweile einige Programme zur Verfügung. Leider arbeiten die meißten lediglich mit Diskette zusammen welche allerdings durch ihren relativ hohen Preis nur eine mäßige Verbreitung verbreitung gefunden hat.
Anders hier bei dem neuen Textprogramm "TEXT 16" von CSJ. Bei diesem Programm wurde nun entlich die Möglichkeit geschaffen, Texte wahlweise auf Diskette oder Kassette abzulegen. Neben dieser Eigenschaft zeigt sich jedoch dieses Programm auch noch in vielen anderer Hinsicht als sehr leistungsstark. So erfolgt beispielsweise die Editierung eines Textes voll bildschirmorientiert. Mit den Cursortasten kann also jede beliebige Stelle des Textes angefahren und Änderungen vorgenommen werden.Gelangt man mittels Cursor über die Bildschirmbegrenzungen, so wird wie bei jedem besseren Textprogramm das Fenster gescrollt. Dies gilt bei dem Programm TEXT 16 sowohl für das überschreiben der horizontalen oder der vertikalen Bildschirmbegrenzung. Etwas mager sieht es mit der maximalen Textgröße aus, lediglich 72 Zeilen zu je 70 Zeichen lassen sicht mit diesem Programm bearbeiten. Für die Korrespondenz dürfte dies zwar ausreichen, wer jedoch mehr als eine DIN A 4 Seite bearbeiten muß oder möchte dürfte an den Grenzen des Programmes scheitern.
Beeindruckent sind allerdings die Anzahl der Funktionen welche in das Programm integriert wurden. Neben Standard Funktionen wie SAVE, LOAD ,LÖSCHEN ,DRUCKEN stehen auch Funktionen zur Verfügung um Textsequenzen zu suchen oder zu ersetzen. Selbst ein WORDWRAP-Algorithmus welcher dafür sorgt das ein Wort welches nicht mehr in die Zeile paßtautomatisch in die nächste übernommen wird, ist in dieses Programm integriert.
Der Aufruf aller Funktionen oder Befehle erfolgt über ein sogenanntes KOMMANDOMENÜ, welches über die HELP Taste eingeblendet wird. Da das Programm somit ausschließlich über ein Menü gesteuert wird, bedarf das Programm keiner langen Einarbeitung, selbst die mitgelieferte Anleitung kann nach wenigen Minuten beiseite gelegt werden.
Im Test zeigte sich das Programm als sehr schnell und benutzerfreundlich. Lediglich der Umstand das Umlaute nicht auf dem Bildschirm zu finden sind ließ Anlaß zur Kritik.
Frank Brall

Positiv:
Voll Bildschirmorientiert
Wordwrap
Zusammenarbeit mit Diskette/Kassette
Gutes Preis/Leistungsverhältnis
Sehr Benutzerfreundlich

Negativ:
Maximal eine DIN A4 Seite berarbeitbar
Keine Umlaute auf dem Bildschirm

Supertool für C-16/116/plus 4

In diesen Tagen soll ein neues Hardwaretool auf den Markt gelangen, daß die Systeme C-16/116/plus 4 um einige wesentliche Funktionen erweitert.
So beinhaltet die Platine, die in den Expansionport gesteckt wird, ein BASICTOOLKIT, welches das eingebaute Basic um die sehr wichtigen Funktionen MERGE, OLD, CHANGE, DUMP und Listing-Scrollen in beide Richtungen erweitert.
Außerdem enthält das Tool noch ein fest eingebautes Turbotape, welches in einer erweiterten Form vorliegt und nun auch die kopletten 64K des plus 4 (oder des aufgerüsteten C-16/116) abspeichern und einladen kann. Das neue Turbotape enthält zudem noch die Möglichkeiten, den MERGE-Befehl im Turbomodus auszuführen und führt zusätzlich noch ein VERIFY aus.
Besonders die plus 4 - Besitzer werden sich über dieses Tool freuen können, da es zwei erhebliche Mankos bzw. Fehler in den eingebauten Programmen des plus 4 beseitigt.
Einmal können jetzt auch Datasettenbesitzer mit dem Textprogramm arbeiten, da beim Laden und Absaven nach der Art des Massenspeichers gefragt wird. Der Programmfehler in der Datenverwaltung wurde auf gleicher Weise behoben. Bei all diesen Vorteilen ist es erstaunlich, daß man für dieses Zubehör nur 49,- DM auf den Ladentisch legen muß.

Info: Kingsoft, F. Schäfer
Schnackebusch 4
5106 Roetgen
Tel.: 02408/5119

```
155 IFK$="5"THEN GOTO 560                                   <115>
160 IFK$="9"THEN PRINT"{CLEAR}":END                         <171>
170 IFK$=CHR$(13)THEN GOTO50                                <135>
180 IFK$=""THEN GOTO50                                      <21>
190 PRINT"{CLEAR}":Q$="":PP=0                               <208>
200 INPUT"{DOWN6 RIGHT2 DOWN}EINGABE{SPACE}[E/A/
Q]:";Q$                                                     <118>
210 PRINT"{CLEAR}"                                          <240>
220 IFQ$="E"THENGOTO460                                     <22>
230 IFQ$="A"THENGOTO510                                     <219>
240 IFQ$=""THENGOTO560                                      <169>
250 IFQ$="Q"THEN GOTO 50                                    <4>
260 PRINT"{CLEAR}"                                          <35>
270 PRINT"{RIGHT14}WERT{SPACE}LOESCHEN"                     <220>
280 PRINT"{RIGHT14 CY13}"                                   <192>
290 INPUT"{DOWN4}BITTE{SPACE}NUMMER{SPACE}DES{SP
ACE}WERTES{SPACE}EINGEBEN{SPACE2}";X                        <85>
300 PRINT"{DOWN}E{SPACE}ODER{SPACE}A":GETKEYEA$             <212>
310 IFEA$="E"THENPRINT"{DOWN2 RIGHT2}";Z(X)                 <173>
320 IFEA$="A"THENPRINT"{DOWN2 RIGHT2}";W(X)                 <101>
330 IFEA$=""THEN 190                                        <34>
340 PRINT"{DOWN2}LOESCHEN{SPACE}ODER{SPACE}AENDE
RN{SPACE}[L/A]{SPACE2}":GETKEYA$:IFA$="L"THEN370            <190>
350 IFA$="A"THEN410                                         <196>
360 GOTO260                                                 <184>
370 H1=0:H2=0                                               <99>
380 IFEA$="E"THEN Z(X)=0.01:H1(X)=1:S(X)=0.00               <216>
390 IFEA$="A"THEN W(X)=0.01:H2(X)=1:D(X)=0.00               <25>
400 GOTO 600                                                <214>
410 INPUT"{DOWN2}NEUER{SPACE}WERT";QW                       <132>
420 FF=(QW*14)/100:DD=(QW/114)*14                           <221>
430 IFEA$="E"THENZ(X)=QW:S(X)=FF                            <150>
440 IFEA$="A"THENW(X)=QW:D(X)=DD                            <250>
450 GOTO600                                                 <8>
460 U=U+1:INPUT"{DOWN RIGHT2}BELEGNUMMER{SPACE2}
";BE1(U):INPUT"{DOWN RIGHT2}EINNAHMEN{SPACE}[DM]
{SPACE2}";Z(U)                                              <152>
470 IFZ(U)=0.00THENGO TO 1690                               <200>
480 S(U)=(Z(U)*14)/100                                      <51>
490 PRINT"{CLEAR}"                                          <9>
500 GO TO 460                                               <108>
510 B=B+1:INPUT"{DOWN RIGHT2}BELEGNUMMER{SPACE2}
";BA1(B):INPUT"{DOWN RIGHT2}AUSGABEN{SPACE2}[DM]
{SPACE2}";W(B)                                              <92>
520 IFW(B)=0.00THEN GO TO 1700                              <40>
530 D(B)=(W(B)/114)*14                                      <41>
540 PRINT"{CLEAR}"                                          <60>
550 GO TO 510                                               <142>
560 INPUT"{CLEAR RIGHT2 DOWN}AUSDRUCK[D/B]";A$              <180>
570 IFA$="D"THENGOTO1760                                    <245>
580 IFA$="B"THENGOTO600                                     <32>
590 IFA$=""THENGOTO50                                       <157>
600 PRINT"{CLEAR RIGHT2}AUSWERTUNG{SPACE}E/A{SPA
CE}RECHNUNG{SPACE2}";TAG$                                   <85>
610 PRINT"{RIGHT2 CY33 DOWN}"                               <35>
620 PRINT"EINNAHMEN{SPACE4}MWST{SPACE5}AUSGABEN{
SPACE3}VORST"                                               <227>
630 PRINT"{CY9 SPACE4 CY4 SPACE5 CY8 SPACE3 CY5}
"                                                           <232>
640 R=0:M=0:O=0:P=0:AS=0:AC=0:AQ=0                          <123>
650 J=0:GOTO1710                                            <62>
660 R=0:M=0:O=0:P=0:AS=0:AC=0:AQ=0                          <143>
670 FORT=1TOG                                               <129>
680 J=J+1:PP=PP+1                                           <38>
690 R=R+Z(T):M=M+S(T):O=O+W(T):P=P+D(T)                     <97>
700 IFZ(T)=0.01THENZ(T)=0.00:R=R-0.01:GOTO720               <43>
710 IFW(T)=0.01THENW(T)=0.00:O=O-0.01:GOTO720               <129>
720 PRINTUSING"#####.##";Z(T)                               <50>
730 PRINTUSING"{UP RIGHT10}####.##";S(T)                    <167>
740 PRINTUSING"{UP RIGHT22}#####.##";W(T)                   <37>
750 PRINTUSING"{UP RIGHT31}####.##";D(T)                    <138>
760 PRINTUSING"{UP RIGHT18 RVSON}##{RVSOFF}";PP             <89>
770 IFJ=18THENJ=0:GETKEYA$:PRINT"{CLEAR}":                  <2>
780 NEXT                                                    <145>
790 PRINT"{DOWN CY9 SPACE CY7 SPACE5 CY8 SPACE C
Y7}"                                                        <96>
800 PRINTUSING"#####.##";R+R1                               <25>
810 PRINTUSING"{UP RIGHT10}####.##";M+M1                    <60>
820 PRINTUSING"{UP RIGHT22}#####.##";O+O1                   <23>
830 PRINTUSING"{UP RIGHT31}####.##";P+P1                    <23>
840 PRINT"{DOWN}=========={SPACE}========{SPACE5}=
========{SPACE}========"                                    <80>
850 AQ=M-P                                                  <185>
860 AS=O-P:AC=R-AS                                          <29>
870 PRINT:PRINT"{SPACE3}GEWINN{SPACE}V.ST.:{SPAC
E2}"USING"{SPACE2}DM{SPACE}########.##";AC                  <238>
880 PRINT:PRINT"{SPACE3}NETTO{SPACE}MWST{RIGHT2}
:{SPACE2}"USING"{SPACE2}DM{SPACE}########.##";AQ            <124>
890 GETKEYA$:GOTO1130                                       <69>
900 PRINT"{CLEAR RIGHT11}SEQ.-{SPACE}FILE{SPACE}
LADEN"                                                      <123>
910 PRINT"{RIGHT11 CY16}"                                   <74>
920 PRINT"{RIGHT7 DOWN2}BITTE{SPACE}BAND{SPACE}R
ICHTIG{SPACE}SPULEN"                                        <98>
930 PRINT"{RIGHT7 DOWN2}DANACH{SPACE}[RETURN]{SP
ACE}DRUECKEN"                                               <240>
940 GETKEYA$:IFA$=CHR$(13)THEN960                           <16>
950 GOTO900                                                 <4>
960 PRINT"{CLEAR}":INPUT"DATEINAME";NAME$                   <59>
970 IFU>BTHENBE1(U+1)=999999:BA1(U+1)=999999:GOT
O990                                                        <207>
980 IFB>UTHENBE1(B+1)=999999:BA1(B+1)=999999                <84>
990 OPEN1,1,O,NAME$                                         <188>
1000 FORT=1TO100                                            <24>
1010 INPUT#1,Z(T)                                           <166>
1020 INPUT#1,S(T)                                           <148>
1030 INPUT#1,W(T)                                           <175>
1040 INPUT#1,D(T)                                           <109>
1050 INPUT#1,BE1(T)                                         <56>
1060 INPUT#1,BA1(T)                                         <46>
1070 INPUT#1,H1(T)                                          <53>
1080 INPUT#1,H2(T)                                          <68>
1090 IFBE1(T)=999999ANDBA1(T)=999999THENCLOSE1:G
OTO1360                                                     <160>
1100 NEXT                                                   <210>
1110 CLOSE 1                                                <92>
1120 GOTO190                                                <186>
1130 GOTO190                                                <196>
1140 PRINT"{CLEAR RIGHT12}SEQ.-{SPACE}FILE{SPACE
}SICHERN"                                                   <158>
1150 PRINT"{RIGHT12 CY16}"                                  <99>
1160 PRINT"{DOWN4 RIGHT8}BITTE{SPACE}BAND{SPACE}
RICHTIG{SPACE}SPULEN"                                       <178>
1170 PRINT"{DOWN2 RIGHT8}DANACH{SPACE}[RETURN]{S
PACE}DRUECKEN"                                              <207>
1180 GETKEYK$:IFK$=CHR$(13)THEN1200                         <49>
1190 GOTO 1140                                              <232>
1200 PRINT"{CLEAR}":INPUT"DATEINAME";NAHME$                 <202>
1210 IFU>BTHENBE1(U+1)=999999:BA1(U+1)=999999               <193>
1220 IFB>UTHENBE1(B+1)=999999:BA1(B+1)=999999               <69>
1230 OPEN1,1,1,NAHME$                                       <57>
1240 FORT=1TO100                                            <9>
1250 PRINT#1,Z(T)                                           <171>
1260 PRINT#1,S(T)                                           <153>
1270 PRINT#1,W(T)                                           <179>
1280 PRINT#1,D(T)                                           <114>
1290 PRINT#1,BE1(T)                                         <61>
1300 PRINT#1,BA1(T)                                         <51>
1310 PRINT#1,H1(T)                                          <58>
1320 PRINT#1,H2(T)                                          <73>
1330 IFBE1(T)=999999ANDBA1(T)=999999THENCLOSE1:G
OTO190                                                      <155>
1340 NEXT                                                   <195>
1350 GOTO90                                                 <214>
1360 PRINT"{CLEAR RIGHT2}AUSWERTUNG{SPACE}E/A{SP
ACE}RECHNUNG{SPACE2}";TAG$                                  <80>
1370 PRINT"{RIGHT2 CY33 DOWN}"                              <30>
1380 PRINT"EINNAHMEN{SPACE4}MWST{SPACE5}AUSGABEN
{SPACE3}VORST"                                              <222>
1390 PRINT"{CY9 SPACE4 CY4 SPACE5 CY8 SPACE3 CY5
}"                                                          <227>
1400 R=0:M=0:O=0:P=0:AS=0:AC=0:AQ=0                         <118>
1410 J=0:GOTO1750                                           <89>
1420 R=0:M=0:O=0:P=0:AS=0:AC=0:AQ=0                         <138>
1430 FORT=1TOG                                              <124>
1440 J=J+1:PP=PP+1                                          <33>
1450 R=R+Z(T):M=M+S(T):O=O+W(T):P=P+D(T)                    <92>
1460 IFZ(T)=0.00ANDH1(T)=1THENU=U+1                         <42>
1470 IFW(T)=0.00ANDH2(T)=1THENB=B+1                         <29>
1480 IFZ(T)>0.00THENU=U+1                                   <246>
1490 IFW(T)>0.00THENB=B+1                                   <12>
1500 IFBE1(T)=999999ANDBA1(T)=999999THENGOTO1580            <213>
1510 PRINTUSING"#####.##";Z(T)                              <75>
1520 PRINTUSING"{UP RIGHT10}####.##";S(T)                   <192>
1530 PRINTUSING"{UP RIGHT22}#####.##";W(T)                  <62>
1540 PRINTUSING"{UP RIGHT31}####.##";D(T)                   <164>
1550 PRINTUSING"{UP RIGHT18 RVSON}##{RVSOFF}";PP            <115>
1560 IFJ=18THENJ=0:GETKEYA$:PRINT"{CLEAR}":                 <27>
1570 NEXT                                                   <170>
1580 PRINT"{DOWN CY9 SPACE CY7 SPACE5 CY8 SPACE
CY7}"                                                       <121>
1590 PRINTUSING"#####.##";R+R1                              <50>
```

```
1600 PRINTUSING"{UP RIGHT10}####.##";M+M1                <85>
1610 PRINTUSING"{UP RIGHT22}#####.##";O+O1               <48>
1620 PRINTUSING"{UP RIGHT31}####.##";P+P1                <48>
1630 PRINT"{DOWN}=========={SPACE}========{SPACE5}
========{SPACE}======="                                  <105>
1640 AQ=M-P                                              <210>
1650 AS=O-P:AC=R-AS                                      <54>
1660 PRINT:PRINT"{SPACE3}GEWINN{SPACE}V.ST.:{SPA
CE2}"USING"{SPACE2}DM{SPACE}########.##";AC              <7>
1670 PRINT:PRINT"{SPACE3}NETTO{SPACE}MWST{RIGHT2
}:{SPACE2}"USING"{SPACE2}DM{SPACE}########.##";A
Q                                                        <149>
1680 GETKEYA$:GOTO40                                     <192>
1690 U=U-1:GOTO190                                       <125>
1700 B=B-1:GOTO190                                       <59>
1710 IFU>BTHENG=U:GOTO1740                               <206>
1720 IFB>UTHENG=B:GOTO1740                               <102>
1730 IFU=BORB=UTHENG=U                                   <228>
1740 GOTO670                                             <44>
1750 G=T:GOTO1430                                        <242>
1760 OPEN 4,4                                            <65>
1770 LI=0                                                <164>
1780 PRINT#4,"{SPACE20}AUSWERTUNG{SPACE}E/A{SPAC
E}RECHNUNG{SPACE2}"+TAG$                                 <214>
1790 PRINT#4,"{SPACE20 CY33}"                            <51>
1800 PRINT#4,""                                          <197>
1810 PRINT#4,"EINNAHMEN{SPACE8}MWST{SPACE8}BENR.
{SPACE5}AUSGABEN{SPACE8}VORST{SPACE8}BENR."              <150>
1820 PRINT#4,"{CY9 SPACE8 CY4 SPACE8 CY5 SPACE5
CY8 SPACE8 CY5 SPACE8 CY5}"                              <63>
1830 FORL=1TOG                                           <254>
1840 LI=LI+1                                             <27>
1850 A1=Z(L)                                             <17>
1860 A2=S(L)                                             <1>
1870 A3=W(L)                                             <29>
1880 A4=D(L)                                             <221>
1890 A5=BE1(L)                                           <154>
1900 A6=BA1(L)                                           <146>
1910 PRINT#4,USING"#####.##{SPACE5}";A1,A2,A5,A3
,A4,A6                                                   <180>
1920 IFLI>60THENGOSUB2020                                <120>
1930 NEXT                                                <19>
1940 PRINT#4,"{CY9 SPACE4 CY8 SPACE18 CY8 SPACE5
 CY8}"                                                   <228>
1950 PRINT#4,USING"#####.##{SPACE5}";R,M,"{SPACE
7}",O,P                                                  <196>
1960 PRINT#4,"=========={SPACE4}========={SPACE18}
========{SPACE5}========"                                <166>
1970 PRINT#4,""                                          <111>
1980 AQ=M-P                                              <39>
1990 PRINT#4,"{SPACE13}NETTO{SPACE}MWST{RIGHT2}:
{SPACE2}"USING"{SPACE2}DM{SPACE}########.##";AQ          <106>
2000 PRINT#4,"{SPACE28}==============="                  <166>
2010 CLOSE4:GOTO40                                       <223>
2020 LI=0:FORT=1TO6:PRINT#4,"":NEXT:RETURN               <60>
  ENDE DES LISTINGS
```

Bildschirmmasken – wirklich »easy«:

Screen-Editor

Dieses Programm ermöglicht es Ihnen, für Ihre eigenen Programme Bildschirmmasken zu entwerfen. Diese können nach Belieben in PRINT-Zeilen eingebettet, als Maschinenprogramm abgespeichert und/oder auf dem Drucker ausgegeben werden. Weiterhin besteht die Möglichkeit, in einer 8*8-Matrix eigene Zeichen zu definieren und diese in die Maske einzubauen.

Nach dem Start von SCREEN-Editor hat man die Whl, ob man die vorgegebenen ASCII-Zeichen verwenden oder eigene Grafikzeichen entwerfen will. Wählt man das Letztere, so hat man folgende Funktionen zur Auswahl:

Cursor-Steuertasten:
Steuerung des Zeichencursors in der Matrix

SPACE:
Punkt setzen bzw. löschen

F:
Neu entworfenes Zeichen speichern

CLEAR/HOME:
Löschen der Matrix

Wenn man keine weiteren Zeichen mehr entwerfen will, tippt man bei der entsprechenden Frage auf die N-Taste und kommt somit in das nächste Menü, welches die Funktionen erläutert, die bei dem Maskenentwurf zur Verfügung stehen.

Die Belegung der Funktionstasten F1, F2, F3 werden im folgenden erklärt:

F1: Nach Drücken dieser Taste wird der Bildschirm gespeichert und es folgt die Frage nach der Zeilennummer, ab der die PRINT-Zeilen beginnen sollen. Nachdem auch dies geklärt ist, beginnt der eigentliche Umwandlungsprozess des entworfenen SCREENS in Programmzeilen. Hiernach erscheint das letzte Menü, daß das Programm bietet. Dieses bedarf keiner weiteren Erläuterung.

F2: Hier wird zuerst auch die Maske gespeichert, bevor der Bildschirm gelöscht und der SCREEN als **Maschinenprogramm** auf Band übertragen wird. Die Maske kann jederzeit wieder mit **LOAD "SCREENNAME",1,1** geladen werden.

F3: Diese Taste sollte nur gedrückt werden, wenn ein Drucker angeschlossen ist. Sollte dies der Fall sein, kann man beruhigt F3 drücken und die bis zu diesem Moment entworfene Maske wird ausgedruckt. Es ist jedoch zu beachten, daß nur der Originalzeichensatz aufs Papier gebracht werden kann. Nach dem Ausdruck können Sie weiter an Ihrem Werk arbeiten oder es mit Hilfe von F1 oder F2 sichern.

Ich denke, daß diese ausführlichen Erläuterungen ausreichen, um Ihnen zu verdeutlichen, welche Möglichkeiten mit diesem Programm gegeben sind. Viel Spaß!

***

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
0 REM  ********************************      <3>
1 REM  *                              *      <98>
2 REM  *        SCREEN - EDITOR       *      <24>
3 REM  *                              *      <100>
4 REM  *                              *      <101>
5 REM  *      BY MARKUS HOLZDEPPE     *      <33>
6 REM  *                              *      <103>
7 REM  *           (C)  1986          *      <58>
8 REM  *                              *      <105>
9 REM  ********************************      <12>
10 POKE65298,196:POKE65299,208                <93>
20 COLOR0,1:COLOR4,1:COLOR1,2:POKE51,0:POKE52,56
:POKE55,0:POKE56,56:CLR                       <213>
30 POKE1344,0                                 <220>
40 PRINT"{CLEAR BLACK}MONITOR{DOWN5}":PRINT"T{SP
ACE}D000{SPACE}D800{SPACE}3800":PRINT"{DOWN}X":P
RINT"{DOWN}GOTO60"                            <209>
50 POKE1319,19:FORI=1320TO1323:POKEI,13:NEXT:POK
E239,5:END                                    <56>
60 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{SPA
CE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{SP
ACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"         <151>
70 PRINT"{DOWN2}BITTE{SPACE}WAEHLEN{SPACE}SIE{SP
ACE}:"                                        <75>
80 PRINT"{DOWN4 RVSON SPACE}1{SPACE RVSOFF SPACE
2}SCREEN{SPACE}MIT{SPACE}DEN{SPACE}VORGEGEBENEN{
SPACE}ZEI{SPACE}-{SPACE5 DOWN SPACE2}CHEN{SPACE}
ENTWERFEN"                                    <120>
90 PRINT"{DOWN3 RVSON SPACE}2{SPACE RVSOFF SPACE
2}NEUE{SPACE}ZEICHEN{SPACE}ENTWERFEN{SPACE}FUER{
SPACE}DEN{SPACE7 DOWN SPACE2}SCREEN"          <119>
100 GETKEYA$:IFA$<>"1"ANDA$<>"2"THEN100       <101>
110 ONVAL(A$)GOTO540,120                      <36>
120 PRINT"{CLEAR}AB{SPACE}WELCHER{SPACE}ZEILE{SP
ACE}SOLLEN{SPACE}DIE{SPACE}DATEN{SPACE}FUER"  <185>
130 INPUT"{DOWN}DIE{SPACE}NEUEN{SPACE}ZEICHEN{SP
ACE}ABGESPEICHERT{SPACE}WERDEN";EZ:IFEZ<20000REZ
>60000THEN120                                 <244>
140 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{SP
ACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{S
PACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"        <231>
150 PRINT"{DOWN}STEUERUNG{SPACE}MIT{SPACE}DEN{SP
ACE RVSON}CURSOR{SPACE}-{SPACE}TASTEN"         <34>
160 PRINT"{DOWN}PUNKTE{SPACE}SETZEN{SPACE}UND{SP
ACE}LOESCHEN{SPACE}MIT{SPACE RVSON SPACE}SPACE{S
PACE}"                                        <253>
170 PRINT"{DOWN}NEUES{SPACE}ZEICHEN{SPACE}SPEICH
ERN{SPACE}MIT{SPACE RVSON SPACE}F{SPACE}"     <145>
180 PRINT"{DOWN}LOESCHEN{SPACE}DES{SPACE}SCREENS
{SPACE}MIT{SPACE RVSON SPACE}CLR{SPACE RVSOFF}"  <108>
190 PRINT"{DOWN2}"                            <188>
200 FORI=1TO8:PRINTTAB(16)".......":NEXT:A=3568
:POKEA,174:X=0:Y=0                            <65>
210 GETKEYA$:IFA$="F"THEN310                  <87>
220 FORT=1TO10:NEXT                           <235>
230 IFA$="{CLEAR}"THENPRINT"{UP9}":GOTO200    <115>
240 IFA$="{LEFT}"ANDX>0THENX=X-1:GOTO300      <109>
250 IFA$="{RIGHT}"ANDX<7THENX=X+1:GOTO300     <200>
260 IFA$="{UP}"ANDY>0THENY=Y-1:GOTO300        <95>
270 IFA$="{DOWN}"ANDY<7THENY=Y+1:GOTO300      <186>
280 IFA$="{SPACE}"ANDPEEK(A)=174THENPOKEA,209:EL
SEPOKEA,174                                   <78>
290 GOTO210                                   <99>
300 POKEA,PEEK(A)-128:A=3568+X+40*Y:POKEA,PEEK(A
)+128:GOTO210                                 <154>
310 FORI=0TO7:W(I)=0:NEXT:P=0:Z=0             <13>
320 POKEA,PEEK(A)-128:FORI=3568TO3848STEP40:FORJ
=1TO1+7                                       <205>
330 IFPEEK(J)=81THENW(Z)=W(Z)+2^(7-P)         <173>
340 P=P+1:NEXT:P=0:Z=Z+1:NEXT                 <11>
350 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{SP
ACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{S
PACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"        <186>
360 POKE65298,192:POKE65299,56:FORI=0TO7:POKE143
36+I,W(I):NEXT                                <184>
370 PRINT"{DOWN}SO{SPACE}SIEHT{SPACE}DAS{SPACE}V
ON{SPACE}IHNEN{SPACE}ENTWORFENE{SPACE}NEUE{SPACE
2 DOWN}ZEICHEN{SPACE}AUS{SPACE}:{SPACE}@"      <3>
380 PRINT"{DOWN2}SOLL{SPACE}ES{SPACE}UEBERNOMMEN
{SPACE}WERDEN{SPACE}?{SPACE}({FLASHON}J{FLASHOFF
}/{FLASHON}N{FLASHOFF})"                      <128>
390 DO:GETKEYA$:LOOPUNTILA$="J"ORA$="N"       <215>
400 IFA$="J"THEN410:ELSE510                   <122>
410 PRINT"{DOWN2}DURCH{SPACE}WELCHES{SPACE}ZEICH
EN{SPACE}SOLL{SPACE}ES{SPACE}ERSETZT{SPACE3 DOWN
}WERDEN{SPACE}?{SPACE}:{SPACE2 LEFT}";        <151>
420 GETKEYA$:PRINTA$:A=PEEK(3563):IFA<10RA>127TH
ENPRINT"{UP6}":GOTO410                        <55>
430 FORI=0TO7:POKE14336+A*8+I,W(I):NEXT       <178>
440 POKE65298,196:POKE65299,208               <12>
450 PRINT"{CLEAR WHITE}"EZ"D{SA}";:PRINTMID$(STR
$(A),2,LEN(STR$(A))-1)",";                    <148>
460 FORI=0TO6:PRINTMID$(STR$(W(I)),2,LEN(STR$(W(
I)))-1)",";:NEXT                              <70>
470 PRINTMID$(STR$(W(7)),2,LEN(STR$(W(7)))-1)  <138>
480 PRINT"EZ="EZ":GOTO500"                    <33>
490 POKE1319,19:POKE1320,13:POKE1321,13:POKE239,
3:END                                         <175>
500 EZ=EZ+1                                   <45>
510 POKE65298,196:POKE65299,208               <82>
520 PRINT"{CLEAR WHITE DOWN2}SOLL{SPACE}EIN{SPAC
E}WEITERES{SPACE}ZEICHEN{SPACE}NEU{SPACE}DEFINIE
RT{SPACE DOWN}WERDEN{SPACE}?{SPACE}({FLASHON}J{F
LASHOFF}/{FLASHON}N{FLASHOFF})"               <225>
530 GETKEYA$:IFA$="J"THEN140                  <214>
540 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{SP
ACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{S
PACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"        <121>
550 PRINT"{DOWN2 RVSON}CURSOR-TASTEN{RVSOFF SPAC
E}:{SPACE}STEUERUNG{SPACE}DES{SPACE}CURSORS"   <133>
560 PRINT"{DOWN RVSON SPACE}C{SPACE}L{SPACE}R{SP
ACE RVSOFF SPACE}:{SPACE}LOESCHEN{SPACE}DES{SPAC
E}SCREENS"                                    <98>
570 PRINT"{DOWN RVSON SPACE}HOME{SPACE RVSOFF SP
ACE}:{SPACE}SETZEN{SPACE}DES{SPACE}CURSORS{SPACE
}AN{SPACE}DIE{SPACE}OBERE{SPACE9}LINKE{SPACE}CUR
SORPOSITION"                                  <172>
580 PRINT"{DOWN RVSON SPACE}D{SPACE}E{SPACE}L{SP
ACE RVSOFF SPACE}:{SPACE}LOESCHEN{SPACE}DES{SPAC
E}ZEICHENS{SPACE}LINKS{SPACE13}DER{SPACE}CURSORP
OSITION"                                      <30>
590 PRINT"{DOWN RVSON SPACE}F{SPACE}1{SPACE RVSO
FF SPACE}:{SPACE}UMWANDLUNG{SPACE}DES{SPACE}SCRE
ENS{SPACE}IN{SPACE RVSON}PRINT{RVSOFF}-{SPACE8}Z
EILEN"                                        <188>
600 PRINT"{DOWN RVSON SPACE}F{SPACE}2{SPACE RVSO
FF SPACE}:{SPACE}ABSPEICHERN{SPACE}DES{SPACE}SCR
EENS{SPACE}ALS{SPACE}MA{SPACE}-{SPACE8}SCHINENPR
OGRAMM"                                       <3>
610 PRINT"{DOWN RVSON SPACE}F{SPACE}3{SPACE RVSO
FF SPACE}:{SPACE}AUSDRUCK{SPACE}DES{SPACE}SCREEN
S{SPACE}(NUR{SPACE}BEI"                       <187>
620 PRINT"{SPACE8}ORIGINALZEICHENSATZ{SPACE}MOEG
LICH{SPACE})"                                 <128>
630 PRINT"{SPACE5 DOWN2}BITTE{SPACE}DRUECKEN{SPA
CE}SIE{SPACE RVSON SPACE}S{SPACE}P{SPACE}A{SPACE
}C{SPACE}E{SPACE RVSOFF}"                     <67>
640 GETKEYA$:IFA$<>"{SPACE}"THEN640           <84>
650 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{SP
ACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{S
PACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"        <231>
660 PRINT"{DOWN2 RVSON SPACE}CTRL{SPACE}+{SPACE}
9{SPACE RVSOFF SPACE}:{SPACE}NEGATIVSCHRIFT{SPAC
E}EIN"                                        <1>
670 PRINT"{DOWN RVSON SPACE}CTRL{SPACE}+{SPACE}0
{SPACE RVSOFF SPACE}:{SPACE}NEGATIVSCHRIFT{SPACE
}AUS"                                         <172>
680 PRINT"{DOWN2}DURCH{SPACE}BETAETIGUNG{SPACE}D
ER{SPACE}ENTSPRECHENDEN"                      <27>
690 PRINT"{DOWN}TASTE{SPACE}ERSCHEINT{SPACE}DAS{
SPACE}GEWUENSCHTE{SPACE}ZEICHEN{SPACE DOWN}AN{SP
ACE}DER{SPACE}CURSORPOSITION."                <59>
700 PRINT"{SPACE5 DOWN5}BITTE{SPACE}DRUECKEN{SPA
CE}SIE{SPACE RVSON SPACE}S{SPACE}P{SPACE}A{SPACE
}C{SPACE}E{SPACE RVSOFF}"                     <174>
710 GETKEYA$:IFA$<>"{SPACE}"THEN710           <121>
720 POKE65298,192:POKE65299,56:KEY1,CHR$(133):KE
Y2,CHR$(137):KEY3,CHR$(134)                   <66>
730 PRINT"{CLEAR}":A=3072:POKEA,160           <159>
740 GETKEYA$                                  <204>
750 IFA$="{UP}"ANDA>3111THENC=-40:GOSUB800:GOTO7
40                                            <230>
760 IFA$="{DOWN}"ANDA<4032THENC=40:GOSUB800:GOTO
740                                           <4>
770 IFA$="{LEFT}"ANDA>3072THENC=-1:GOSUB800:GOTO
740                                           <221>
780 IFA$="{RIGHT}"ANDA<4071THENC=1:GOSUB800:GOTO
740                                           <4>
790 GOTO830                                   <107>
800 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKEA
,PEEK(A)+128                                  <89>
810 A=A+C:IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELS
```

```
EPOKEA,PEEK(A)+128                                                  <192>
820 RETURN                                                          <197>
830 IFA$="{CLEAR}"THEN730                                           <193>
840 IFA$="{HOME}"THENC=-(A-3072):GOSUB800:GOTO74
0                                                                   <106>
850 IFA$="{F1}"THEN1000                                             <109>
860 IFA$="{F2}"THEN1330                                             <206>
870 IFA$="{F3}"THEN1450                                             <230>
880 AS=ASC(A$):IFAS>63ANDAS<96THENPO=AS-64:GOTO9
70                                                                  <103>
890 IFAS>191ANDAS<224THENPO=AS-128:GOTO970                          <52>
900 IFAS>31ANDAS<64THENPO=AS:GOTO970                                <110>
910 IFAS=255THENPO=94:GOTO970                                       <232>
920 IFAS>159ANDAS<192THENPO=AS-64:GOTO970                           <158>
930 IFAS=18THENR=1:GOTO740                                          <98>
940 IFAS=146THENR=0:GOTO740                                         <253>
950 IFAS=20ANDA>3072ANDPEEK(A)=160THENPOKEA,32:A
=A-1:POKEA,160:GOTO740                                              <189>
960 GOTO740                                                         <22>
970 POKEA,PO+128*R:A=A+1:IFA=4072THENA=3072                         <61>
980 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKEA
,PEEK(A)+128                                                        <13>
990 GOTO740                                                         <52>
1000 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKE
A,PEEK(A)+128                                                       <33>
1010 COLOR4,8:FORI=0TO999:POKE15360+I,PEEK(3072+
I):NEXT:COLOR4,1                                                    <43>
1020 POKE65298,196:POKE65299,208                                    <82>
1030 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{S
PACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{
SPACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"                            <101>
1040 PRINT"{DOWN2}AB{SPACE}WELCHER{SPACE}ZEILE{S
PACE}SOLLEN{SPACE}DIE{SPACE RVSON}PRINT{RVSOFF}-
ZEILEN":INPUT"{DOWN}BEGINNEN";EZ                                    <248>
1050 IFEZ<2000OREZ>60000THEN1030                                    <209>
1060 I=15360                                                        <207>
1070 PRINT"{CLEAR}"EZ"PRINT"CHR$(34);                               <20>
1080 FORJ=ITOI+39:PO=PEEK(J)                                        <115>
1090 IFPO>127ANDR=0THENPRINTCHR$(34)"CHR$(18)"CH
R$(34);:R=1                                                         <46>
1100 IFPO<128ANDR=1THENR=0:PRINTCHR$(34)"CHR$(14
6)"CHR$(34);                                                        <141>
1110 IFPO>127THENPO=PO-128                                          <112>
1120 IFPO<32THENAS=PO+64:GOTO1160                                   <167>
1130 IFPO>31ANDPO<64THENAS=PO:GOTO1160                              <186>
1140 IFPO>63ANDPO<96THENAS=PO+32:GOTO1160                           <109>
1150 IFPO>95ANDPO<128THENAS=PO+64                                   <176>
1160 PRINTCHR$(AS);:NEXT:PRINTCHR$(34):PRINT"EZ=
"EZ":I="I":GOTO1180"                                                <13>
1170 POKE239,3:POKE1319,19:POKE1320,13:POKE1321,
13:END                                                              <60>
1180 EZ=EZ+1:I=I+40:IFI<=16320THEN1070                              <182>
1190 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{S
PACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{
SPACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"                            <5>
1200 PRINT"{DOWN2}BITTE{SPACE}WAEHLEN{SPACE}SIE{
SPACE}JETZT{SPACE}:{SPACE15 S*25}"                                  <35>
1210 PRINT"{DOWN2 RVSON SPACE}1{SPACE RVSOFF SPA
CE2}NEUER{SPACE}SCREEN"                                             <95>
1220 PRINT"{DOWN2 RVSON SPACE}2{SPACE RVSOFF SPA
CE2}SCREEN{SPACE}AUSDRUCKEN"                                        <44>
1230 PRINT"{DOWN2 RVSON SPACE}3{SPACE RVSOFF SPA
CE2}PROGRAMMENDE"                                                   <99>
1240 GETKEYA$:IFA$<"1"ORA$>"3"THEN1240                              <240>
1250 ONVAL(A$)GOTO1270,1260,1610                                    <253>
1260 FORI=0TO999:POKEI+3072,PEEK(I+15360):NEXT:G
OSUB1470:GOTO1190                                                   <170>
1270 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{S
PACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{
SPACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"                            <85>
1280 PRINT"{DOWN2}BITTE{SPACE}WAEHLEN{SPACE}SIE{
SPACE}JETZT{SPACE}:{SPACE15 S*25}"                                  <116>
1290 PRINT"{DOWN2 RVSON SPACE}1{SPACE RVSOFF SPA
CE2}NEU{SPACE}DEF.{SPACE}ZEICHENSATZ{SPACE}LOESC
HEN"                                                                <115>
1300 PRINT"{DOWN4 RVSON SPACE}2{SPACE RVSOFF SPA
CE2}NEU{SPACE}DEF.{SPACE}ZEICHENSATZ{SPACE}BEIBE
HALTEN"                                                             <157>
1310 GETKEYA$:IFA$<"1"ORA$>"2"THEN1310                              <246>
1320 ONVAL(A$)GOTO40,60                                             <199>
1330 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKE
A,PEEK(A)+128                                                       <109>
1340 COLOR4,8:FORI=0TO999:POKE15360+I,PEEK(3072+
I):NEXT:COLOR4,1                                                    <119>
1350 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{S
PACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{
SPACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"                            <166>
1360 INPUT"{DOWN2}SCREENNAME";N$:IFLEN(N$)<10RLE
N(N$)>16THENPRINT"{UP4}":GOTO1360                                   <124>
1370 AA=3072:EA=4072:PRINT"{DOWN2 SPACE7 RVSON}P
LAY{RVSOFF SPACE}+{SPACE RVSON}RECORD{RVSOFF SPA
CE}DRUECKEN"                                                        <199>
1380 IFPEEK(1)<>192THEN1380                                         <247>
1390 FORI=0TO999:POKE3072+I,PEEK(15360+I):NEXT                      <133>
1400 SYS43115 N$,1,1                                                <251>
1410 H1=INT(AA/256):POKE217,H1:L1=AA-256*H1:POKE
216,L1                                                              <253>
1420 H2=INT(EA/256):POKE2036,H2:L2=EA-256*H2:POK
E2035,L2                                                            <225>
1430 POKE2034,216:SYS65496                                          <160>
1440 POKE65298,196:POKE65299,208:GOTO1190                           <126>
1450 GOSUB1460:GOTO1590                                             <204>
1460 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKE
A,PEEK(A)+128                                                       <239>
1470 OPEN4,4:FORI=3072TO4032STEP40                                  <49>
1480 PRINT#4,CHR$(15)CHR$(16)"20";                                  <143>
1490 FORJ=ITOI+39:PO=PEEK(J)                                        <14>
1500 IFPO>127ANDR=0THENPRINT#4,CHR$(18);:R=1                        <69>
1510 IFPO<128ANDR=1THENR=0:PRINT#4,CHR$(146);                       <212>
1520 IFPO>127THENPO=PO-128                                          <11>
1530 IFPO<32THENAS=PO+64:GOTO1570                                   <162>
1540 IFPO>31ANDPO<64THENAS=PO:GOTO1570                              <197>
1550 IFPO>63ANDPO<96THENAS=PO+32:GOTO1570                           <135>
1560 IFPO>95ANDPO<128THENAS=PO+64                                   <76>
1570 PRINT#4,CHR$(AS);:NEXT:PRINT#4,CHR$(B):NEXT
:PRINT#4,CHR$(15):PRINT#4:CLOSE4                                    <184>
1580 RETURN                                                         <192>
1590 IFPEEK(A)>127THENPOKEA,PEEK(A)-128:ELSEPOKE
A,PEEK(A)+128                                                       <114>
1600 GOTO740                                                        <153>
1610 PRINT"{WHITE CLEAR RVSON SPACE7}S{SPACE}C{S
PACE}R{SPACE}E{SPACE}E{SPACE}N{SPACE3}E{SPACE}D{
SPACE}I{SPACE}T{SPACE}O{SPACE}R{SPACE8}"                            <171>
1620 PRINT"{DOWN4 SPACE2}SOLL{SPACE}DAS{SPACE}PR
OGRAMM{SPACE}GELOESCHT{SPACE}WERDEN{SPACE}?"                        <241>
1630 PRINT"{DOWN SPACE16 RVSON SPACE}({FLASHON}J
{FLASHOFF}/{FLASHON}N{FLASHOFF}){SPACE}"                            <84>
1640 GETKEYA$:IFA$="N"THEN1680                                      <42>
1650 IFA$<>"J"THEN1640                                              <15>
1660 PRINT"{CLEAR}DELETE-1680":PRINT"{DOWN2}PRIN
TCHR$(147):END"                                                     <207>
1670 POKE1319,19:POKE1320,13:POKE1321,13:POKE239
,3:END                                                              <80>
1680 PRINT"{CLEAR}":END                                             <214>
  ENDE DES LISTINGS
```

»dload« erweitert!

Das Basic 3.5 kennt den Befehl "dload" sowohl im Commando-, wie auch im Befehlsmodus.
Im Befehlsmodus (also als Aufruf aus einem Programm heraus) hat dieser Befehl jedoch einen Haken:
Das aufzurufende Programm muss kürzer sein, als das aufrufende Programm. Und zwar nicht nur, wie im Handbuch erwähnt, um die Variablen zu erhalten, sondern vor allem, weil das nachgeladene Programm sonst nicht vollständig geladen wird.

Es gibt jedoch eine einfache Möglichkeit der Abhilfe, indem man den Computer den Commandomodus vorgaukelt. Dies geschieht mit Hilfe des Tastaturpuffers. Als kurze Erläuterung so viel: Der Bildschirm wird gelöscht, damit eventuell vorhandene Zeichen

nicht zur ERROR-Meldung führen. Dann wird "dload" und der Programmname mit einem Printbefehl auf den Bildschirm geschrieben. Die nachfolgende READY-Meldung des Rechners führt dazu, daß der Cursor zwei Zeilen zu tief steht. Durch zweimaligen Cursor-up-Befehl (chr$ (145)), wird der Cursor wieder in die richtige Zeile gebracht. Im Tastaturpuffer stehen nun "RETURN" und anschließend "RUN" und "RETURN". Das heißt, Das Programm wird nachgeladen und gestartet, wie dies eigentlich mit dem schlichten "dload" auch funktionieen sollte. Und nun das kurze Programm:

```
10
scnclr
20 a$='Programmname'
30 print'dload'
40 print chr$(34);a$;
50 poke 1319,145
60 poke 1320,145
70 poke 1321,13
80 poke 1322,82
90 poke 1323,85
100 poke 1324,78
110 poke 1325,13
120 poke 239,7
```

Dieses Utility als Unterprogramm eingesetzt, ermöglicht zum Beispiel die Erstellung eines Diskettenmenüs: sämtliche Programme einer Diskette werden von einem Menü aus nachgeladen.

Na endlich: Speichersplit auch für die 64-K-Versionen

Speichersplit für Plus/4 und C16 + 64 KByte

Speichersplit teilt den Arbeitsspeicher in zwei selbständige Bereiche auf, in welche je ein Basicprogramm geladen, und beide unabhängig voneinander bearbeitet werden können. Mit Maschinenprogrammen ist Speichersplit nur dann kompatibel, wenn keine Überschneidungen vorkommen (vgl. Tabelle).

So kann man beispielsweise, während manim Bereich 0 ein Programm erstellt, ein zweites im Bereich 1 bereithalten und Teile daraus mittels des Bildschirmeditors in den Bereich 0 übernehmen. Das spart Arbeit und Zeit. Oder man hält, während man im einen Bereich Vokabeln paukt, im anderen ein kleines Spielchen bereit, um zur Entspannung hin und wieder mal umzuschalten. Dabei kann auch die hochauflösende Grafik uneingeschränkt benutzt werden.

Das Hin- und Herschalten zwischen den beiden Speicherblöcken geschieht mittels der Funktionstaste F8. (HELP). Nach jedem HELP Tastendruck meldet der Computer in welchem Bereich man sich nun befindet und wieviele Bytes hier noch frei sind. Alle Basickommandos (RUN, LIST, SAVE, LOAD, NEW etc.) beziehen sich ausschließlich auf den aktuellen Bereich.

Als zusätzliche Hilfe sind die Funktionstasten F1 und F2 mit der Funktion OLD für die Bereiche 0 bzw. 1 belegt. Damit kann ein versehentlich gegebenes NEW wieder rückgängig gemacht werden.

Taste F6 führt einen RESET durch.

Nach dem Eintippen des Basicloaders ist es ganz wichtig diesen zuerst abzuspeichern, denn er löscht sich selbst, nachdem das Maschinenprogramm generiert worden ist.

REM's können selbstverständlich weggelassen werden.

***

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
10 REM *** (C) BERNHARD LAUER, 1986 ***                <15>
20 :                                                   <78>
30 POKE216,0 : REM BEREICHSZEIGER DB                   <247>
40 :                                                   <98>
50 REM GRAFIKBEREICH RESERVIEREN                       <125>
60 GRAPHIC 1 :GRAPHIC 0                                <146>
70 :                                                   <128>
80 REM PLATZ SCHAFFEN                                  <165>
90 KEY3,"DI{SR}"+CHR$(13):KEY 4,"":KEY 5,""            <183>
100 :                                                  <158>
110 REM FUNKTIONSTASTENBELEGUNG                        <19>
120 :                                                  <178>
130 REM OLD FUER SPEICHERBER.1 = F1                    <113>
140 KEY1,"P{SO}16386,1:S{SY}34840:S{SY}34892"+CH
R$(13)                                                 <189>
150 :                                                  <208>
160 REM OLD FUER SPEICHERBER.2 = F2                    <188>
170 KEY2,"P{SO}45058,1:S{SY}34840:S{SY}34892"+CH
R$(13)                                                 <199>
180 :                                                  <238>
190 REM RESET    = F6                                  <30>
200 KEY6,"S{SY}32793"+CHR$(13)                         <6>
210 :                                                  <12>
220 REM UMSCHALTEN ZWISCHEN DEN SPEICHER    BERE
ICHEN = F8 (HELP)                                      <39>
230 KEY 8,"S{SY}1630:C{SL}:P{SO}194,1:?P{SE}(216
),F{SR}(X)"+CHR$(13)                                   <51>
240 :                                                  <42>
250 REM SPEICHERGRENZE BEREICH 0 SETZEN                <225>
260 POKE52,175:POKE54,175:POKE56,175                   <106>
270 :                                                  <73>
280 REM ANFANGSWERTE FUER ZEIGERSPEICHER555 :          <151>
290 FOR I=0TO13                                        <203>
300 READ A$                                            <162>
310 POKE 1694+I,DEC(A$)                                <142>
320 NEXT I                                             <85>
330 :                                                  <133>
340 REM NULLBYTES AM BEGINN DER BEREICHE               <253>
350 POKE 45056,0:POKE16384,0                           <244>
360 FOR I=1 TO 10 :POKE 45056+I,0:NEXT                 <155>
370 :                                                  <173>
380 REM MASCHINENPROGRAMM                              <46>
390 FOR I=1630 TO 1669 :READ A:POKE I,A:S=S+A:NE
XT                                                     <111>
400 IF S<>5431 THEN PRINT"FEHLER{SPACE}IN{SPACE}
DATAS!:END                                             <200>
410 NEW                                                <61>
420 :                                                  <223>
430 DATA 01,B0,03,B0,03,B0,03                          <26>
440 DATA B0,FF,FC,FD,FC,FF,FC                          <53>
450 :                                                  <253>
460 DATA162,014,181,043,157,174,006,202                <153>
470 DATA208,248,162,014,189,158,006,149                <150>
480 DATA043,202,208,248,234,162,014,189                <22>
490 DATA174,006,157,158,006,202,208,247                <167>
500 DATA165,216,073,001,133,216,096,000                <77>
  ENDE DES LISTINGS
```

Hier ist es:

Unser Turbo-Tape!

120 "TURBO"
180 "GRAPHIC"
250 - 257 F1

Wer hat sich nicht schon mal die mangelnde Geschwindigkeit der C-16'er Datasette geärgert?

Turbo-Tape schafft hier Abhilfe!

Turbo-Tape speichert Programme nicht nur wesentlich schneller ab, es ermöglicht im Gegensatz zu anderen Schnelladeprogrammen auch das Laden in den "kalten" Rechner, also ohne Turbo-Tape.

Weiterhin ist eine Autostartroutine eingebaut, die es auf Wunsch ermöglicht, Programme nach dem Laden sofort automatich zu starten.

Nach dem Laden und Starten von TURBO-TAPE erscheint auf dem Bildschirm ein Titelbild, in dem unter anderem der noch verbleibende Speicherplatz angezeigt wird. Da TURBO-TAPE die ersten 1228 Bytes des Basicbereichs belegt, ist es auch in der 16K-Version weiterhin möglich, hochauflösende Grafik uneingeschränkt zu benutzen.

Die **Steuerung** erfolgt über 3 Befehle:

>SAVE
>LOAD
>VERIFY.

Abspeichern:
>SAVE speichert das Programm im Basic-Speicher in TURBO-TAPE-Geschwindigkeit ab. Wird hinter dem Befehl ein File-Name in Anführungszeichen oder in einer Stringvariablen geschrieben, so wird das Programm unter diesem Namen abgespeichert. Es werden dabei nur die ersten 17 Zeichen benutzt, der Rest wird "abgeschnitten".

Nachdem der Befehl abgeschickt wurde, erscheint auf dem Bildschirm der Text **"turbo-saving"** und die Aufforderung **"press play & record on tape"**. Sobald man die Recordtaste an der Datasette drückt, wird der Bildschirm leer und der Recorder läuft an. Nach wenigen Sekunden erscheinen schwarze Balken auf dem Bildschirm. Dies ist ein Zeichen dafür, daß TURBO-TAPE aktiv ist. Nachdem das Programm abgespeichert ist, erscheint eine READY-Meldung.

Abgespeicherte Programme solltn immer mit >VERIFY überprüft werden:
>VERIFY überprüft byteweise das im Speicher befindliche Programm mit dem auf der Kassette. Dies geschieht ähnlich der VERIFY-Routine des Betriebssystems. Daher ist es auch möglich, über die >VERIFY-Routine das Ende eines Programmes auf Kassette zu finden. Dadurch kann man Programme direkt hintereinander ohne die Gefahr eines Überschreibens abspeichern. (**Achtung:** bei einem TURBO-Programm funktioniert dies nur mit >VERIFY, während ein normal abgespeichertes Programm auch mit dem neuen Befehl überprüft werden kann.)
Parameter hinter VERIFY werden ignoriert.

Die >VERIFY-Funktion kann 3 verschiedene Meldungen ausgeben:

1. FILE OK.
   Das Programm auf Kassette ist korrekt abgespeichert und identisch mit im dem Speicher.
2. ?VORSPANN:
   VERIFY ERROR
   Der Programmvorspann, der in normaler Form abgespeichert wurde, ist fehlerhaft. Dies weist auf schwere Übertragungsfehler hin. Dieser Fehler tritt auch auf, wenn ein Turbo-Tape-fremdes File vorlag.
3. ?VERIFY ERROR
   Das Programm in TURBO-TAPE-Geschwindigkeit ist entweder nicht mit dem im Speicher identisch oder es wurde fehlerhaft übertragen.

LADEN:
>LOAD entspricht dem normalen LOAD-Befehl. Es wird hier lediglich das erste Zeichen (">")ignoriert. Dieser Befehl wurde nur der vollständigkit halber eingebaut.
Um Programme in TURBO-TAPE-Geschwindigkeit zu laden, ist das TURBOØTAPE-Programm selbst nicht notwendig. Daher kann man solche Programme auch in den "kalten" Computer laden, das heißt: direkt

nach dem Einschalten. Dazu muß man nur LOAD eingeben. Wenn beim Laden ein Fehler auftritt, wird eine
LOAD ERROR - Meldung ausgegeben.

Autostart:
Eine der wesentlichen Vorteile von TURBO-TAPE C 16 liegt in der Fähigkeit zum Autostart.
Soll das Basicprogramm nach dem Laden direkt gestartet werden, so muß man beim Abspeichern einen Stern (*) als erstes Zeichen in den Filenamen setzen. Nach dem Laden wird dann das Programm direkt durch RUN gestartet.

BREAK-Möglichkeiten:
Es gibt verschiedene Möglichkeiten, ein Programm anzuhalten. Zunächst ist es jederzeit möglich die RESET-Tast bei gedrückter RUN-STOP-Taste in den Monitor zu gelangen. Ansonsten kann man noch bei gedrückter RUN-STOP-Taste die STOP-Taste der Datasette betätigen, was allerdings bei >SAVE nicht funktioniert, solange die schwarzen Balken zu sehen sind.

Störungen:
Trotz der hohen Übertragungsgeschwindigkeit (TURBO-TAPE: 1500 Baud, normal 250 Baud) konnte eine sehr hohe Sicherheit erreicht werden. Dennoch sollten einige Regeln beachtet werden:

- Es sollten nur qualitativ hochwertige Normalkassetten (keine Chromdioxid) verwendet werden.
- Um Störungen zu vermeiden, sollte der Abstand der Datasette zu Computer, Fernseher und Netzgeräten mindestens ein Meter betragen.
- Der Tonkopf sollte regelmäßig gereinigt werden.

Sollten dennoch Störungen auftreten, so kann dies an der Magnetisierung des Tonkopfes, an einer schlechten Justierung oder schlechten Erdung liegen.

Tips zum Eingeben:
Bevor man mit dem Eintippen beginnt, muß man folgende Zeile im Direktmodus eingeben:
POKE 43,1 : POKE 8192,0 : NEW

Danach kann das Programm abgetippt und gespeichert werden. Beim Starten werden die Datazeilen gelesen und geprüft. Sollten Fehler gefunden werden, so werden sie angezeigt. Wenn all Datazeilen korrekt waren, gibt das Programm die Meldung "DATAS OK" aus und setzt die Basiczeiger auf das neue Programm.
Listet man jetzt, sieht man folgendes kleine Basicprogramm:
1986 SYS 4128:REM TURBO-TAPE

Man kann es nun über SAVE abspeichern. Wenn man zum Abspeichern die F1-Taste benutzt, wird das Programm automatisch in TURBO-TAPE-Form abgespeichert und mit einer Autostartroutine versehen.
Es muß dann nur noch LOAD eingegeben werden, um das Programm zu laden und zu starten. Nach dem Abspeichern sollte unbedingt RUN eingegeben werden.
Wurde das TURBO durch ein RESET gelöscht, kann es durch **SYS 4128** reaktiviert werden.

Arbeitsweise:
Turbotape belegt den Anfang des Basicspeichers. der Basicanfang wurde entsprechend verschoben. Zum Abspeichern wird eine Laderoutine in Eingabepuffer, Stack und Programmname gelegt und abgespeichert. Der Programmtext wird dann in einem 3-Ton Verfahren bitweise abgespeichert, mit einem Prüfbit pro Byte und einer 16-Bit Prüfsumme am Ende.

Übertragungszeiten:

Länge	normal	TURBO
12K	6'50	1'15
50K	27'00	4'28

C-16 LISTING + CHECKSUMMEN (DC V1.0)

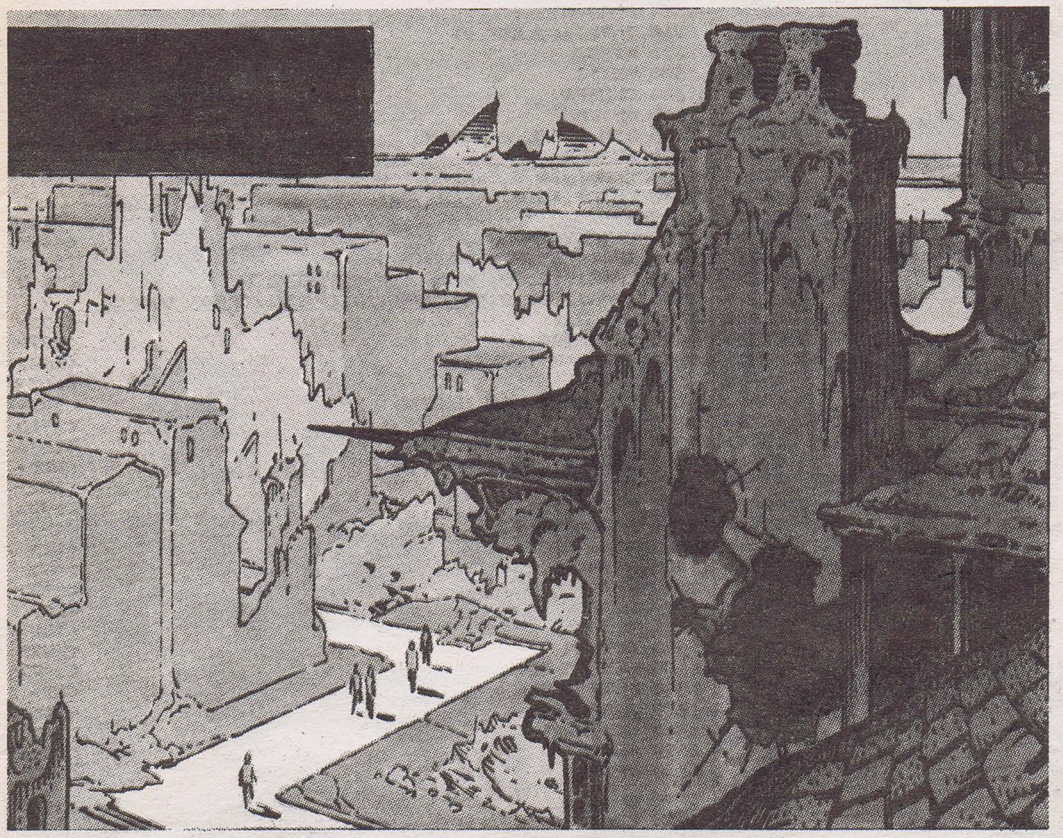
```
0 REM ********************************      <3>
1 REM ***                              ***  <254>
2 REM *** T U R B O - T A P E   C16 ***  <239>
3 REM ***                              ***  <0>
4 REM ***   (C)1986 FROG-SOFTWARE      ***  <124>
5 REM ***                              ***  <2>
6 REM ***   MADE BY JOACHIM THEES      ***  <227>
7 REM ***           GUTENTHAL 32B      ***  <159>
8 REM ***           5552  MORBACH      ***  <81>
9 REM ********************************      <12>
10 :                                        <68>
20 PRINTCHR$(27)+"N"                        <76>
30 IF PEEK(44)=32 THEN 130                  <28>
40 :                                        <98>
50 REM POKES VERGESSEN !                    <105>
60 PRINT"?FEHLER:POKES{SPACE}NICHT{SPACE}EINGEGE
BEN{SPACE}!"                                <54>
70 PRINT:PRINT"{SPACE8}GEBEN{SPACE}SIE{SPACE}FOL
GENDE{SPACE}POKES{SPACE}EIN"                <21>
80 PRINT:PRINT"{SPACE8}UND{SPACE}LADEN{SPACE}SIE
{SPACE}DAS{SPACE}PROGRAMM{SPACE}NEU."       <197>
90 PRINT                                    <243>
100 PRINT"{SPACE}POKE{SPACE}43,1:POKE{SPACE}44,3
2:POKE{SPACE}8192,0:NEW":PRINT              <104>
110 STOP                                    <254>
120 :                                       <178>
130 REM POKES OK.                           <52>
140 PRINT"{SPACE RIGHT SPACE4}T{SPACE}U{SPACE}R{
SPACE}B{SPACE}O{SPACE}-{SPACE}T{SPACE}A{SPACE}P{
SPACE}E{SPACE2}V{SPACE}2.0"                 <16>
150 PRINT"{SPACE RIGHT SPACE4 CY26}"        <198>
160 PRINT"{DOWN2}BITTE{SPACE}WARTEN{SPACE}DATAS{
SPACE}WERDEN{SPACE}GELESEN.":PRINT          <65>
170 :                                       <228>
180 FORBL=0TO15                             <44>
190 PRINT"BLOCK{SPACE}#";BL;                <1>
200 SU=0:LE=79:IFBL=15THENLE=31             <209>
210 FORBB=0TOLE                             <105>
220 READ BY:PA=4096+BL*80+BB:SU=SU+BY       <25>
230 POKE PA,BY                              <240>
240 NEXT BB                                 <188>
250 READ SS                                 <32>
260 IF SS=SU THEN 300                       <77>
270 REM FEHLER                              <113>
280 PRINT"...FEHLER{SPACE}!":PRINT          <200>
290 STOP                                    <179>
300 PRINT"{SPACE}OK."                       <31>
310 NEXT BL                                 <33>
320 PRINT"{DOWN2}DATAS{SPACE}OK."           <9>
330 A$="RUN"+CHR$(13)+"POKE43,1:POKE44,16:POKE45
,205:POKE46,20"+CHR$(13)                    <181>
340 A$=A$+">SAVE"+CHR$(34)+"*TURBO-TAPE{SPACE}V2
.0"+CHR$(13)                                <172>
350 KEY1,A$                                 <246>
360 PRINT:PRINT"PROGRAMM{SPACE}FERTIG.{SPACE}ABS
PEICHERN{SPACE}UEBER"                       <84>
370 PRINT:PRINT"{SPACE RVSON}SAVE{RVSOFF SPACE}O
DER{SPACE RVSON SPACE}F1{SPACE RVSOFF}!"    <25>
380 PRINT:PRINT                             <85>
390 POKE43,1:POKE44,16:POKE45,205:POKE46,20:CLR:
END                                         <252>
400 :                                       <203>
1000 :                                      <37>
1010 REM PROGRAMM-BLOCK # 0                 <35>
1020 DATA   0, 24, 16,194,  7,158, 52, 49   <63>
1030 DATA  50, 56, 58,143, 32, 84, 85, 82   <27>
1040 DATA  66, 79, 45, 84, 65, 80, 69,  0   <99>
1050 DATA   0,  0,  0,  0,  0,  0,  0,  0   <245>
1060 DATA  76,101, 17, 68, 65, 84, 65, 71   <69>
1070 DATA  69, 78, 43, 83, 85, 77, 32, 32   <21>
1080 DATA  32, 32, 32, 32, 32, 32,100,227   <68>
1090 DATA  32,141,227,162,  0,134,212,134   <161>
1100 DATA 213,166, 43,164, 44,134,208,132   <86>
1110 DATA 209, 32, 56,  2, 32, 56,  2,176   <186>
1120 DATA 5854                              <209>
1130 :                                      <168>
1140 REM PROGRAMM-BLOCK # 1                 <183>
1150 DATA   3, 32, 31,  2, 32, 56,  2,176   <142>
1160 DATA 251, 32, 53,  2, 32,  8,  2,160   <92>
1170 DATA   0,145,208, 24,101,212,133,212   <129>
1180 DATA 144,  2,230,213,230,208,208,  2   <110>
1190 DATA 230,209, 32, 53,  2,144,229, 32   <193>
1200 DATA   8,  2, 69,212,208, 34, 32,  8   <75>
1210 DATA   2, 69,213,208, 27, 32,  8,  2   <25>
1220 DATA 162,  0,200,208,253,232,208,250   <90>
1230 DATA 165,  1,  9,  8,133,  1, 56,110   <191>
1240 DATA 252,  7, 32,120,227, 76,188,  3   <143>
1250 DATA 8112                              <42>
1260 :                                      <42>
1270 REM PROGRAMM-BLOCK # 2                 <74>
1280 DATA  32,120,227,162, 29, 76,131,134   <116>
1290 DATA 162,248,154,169,139, 72,169,219   <89>
1300 DATA  72,162,  5, 36,129, 48, 18,173   <23>
1310 DATA  55,  3,201, 42,208, 11,134,239   <72>
```

```
1320 DATA 189,227,  3,157, 38,  5,202,208   <158>
1330 DATA 247,166,208,164,209, 76, 48,168   <124>
1340 DATA  82, 85, 78, 58, 13,234,234,234   <2>
1350 DATA 234,234, 76, 73,  3,162,  8,134   <152>
1360 DATA 216, 72,173, 25,255, 73,127,141   <235>
1370 DATA  25,255, 32, 53,  2,104,106,198   <243>
1380 DATA 9893                              <233>
1390 :                                      <173>
1400 REM PROGRAMM-BLOCK # 3                 <222>
1410 DATA 216,208,238, 96,169, 16, 36,  1   <140>
1420 DATA 208,252,160,255, 36,  1,240,252   <97>
1430 DATA 174,  2,255,140,  2,255,140,  3   <211>
1440 DATA 255, 96, 32, 31,  2, 32, 31,  2   <23>
1450 DATA 202,134,  3,173, 16,253, 41,  4   <63>
1460 DATA 240,  8, 32,120,227,162, 30, 76   <236>
1470 DATA 131,134, 32, 31,  2,228,  3, 96   <219>
1480 DATA 162,255,134, 58, 32, 90,136,134   <103>
1490 DATA  59,132, 60, 32,115,  4,170,240   <69>
1500 DATA 239,176,  3, 76, 46,135, 32, 83   <153>
1510 DATA 8817                              <93>
1520 :                                      <47>
1530 REM PROGRAMM-BLOCK # 4                 <113>
1540 DATA 137, 32,121,  4,201,177,240,  3   <124>
1550 DATA  76, 40,135, 32,115,  4,201,147   <197>
1560 DATA 240,246,201,148,208,  3, 76,140   <90>
1570 DATA  18,201,149,208,  3, 76,130, 19   <164>
1580 DATA 162, 11, 76,131,134,162,205,160   <175>
1590 DATA  20,134, 43,132, 44, 32,123,138   <155>
1600 DATA 162, 40,160, 17,142,  2,  3,140   <19>
1610 DATA   3,  3,162,214,160, 17,134,214   <34>
1620 DATA 132,215, 32,129,255, 32,132,255   <139>
1630 DATA  32, 79,255, 17, 84, 82, 79, 78   <153>
1640 DATA 8899                              <10>
1650 :                                      <178>
1660 REM PROGRAMM-BLOCK # 5                 <5>
1670 DATA  73, 67, 32, 84, 85, 82, 66, 79   <154>
1680 DATA  45, 84, 65, 80, 69, 32, 86, 50   <140>
1690 DATA  46, 48, 32, 32,  0,165, 55, 56   <44>
1700 DATA 229, 43,170,165, 56,229, 44, 32   <229>
1710 DATA  95,164, 32, 79,255, 32, 66, 89   <137>
1720 DATA  84, 69, 83, 32, 70, 82, 69, 69   <142>
1730 DATA   0,160,  0,177,214,240, 12, 32   <222>
1740 DATA 210,255,230,214,208,243,230,215   <91>
1750 DATA  76,193, 17, 76,  3,128, 13, 13   <180>
1760 DATA  13, 13, 32, 32, 65, 76, 76, 32   <147>
1770 DATA 7349                              <103>
1780 :                                      <52>
1790 REM PROGRAMM-BLOCK # 6                 <152>
1800 DATA  86, 69, 82, 83, 73, 79, 78, 83   <86>
1810 DATA  32, 49, 54, 46, 46, 54, 52, 32   <136>
1820 DATA  75, 66, 32, 79, 78, 32, 67, 49   <252>
1830 DATA  54, 47, 49, 49, 54, 47, 43, 52   <39>
1840 DATA  13, 13, 32, 32, 32, 32, 32, 32   <93>
1850 DATA  32, 32, 32, 40, 67, 41, 49, 57   <32>
1860 DATA  56, 54, 32, 70, 82, 79, 71, 45   <82>
1870 DATA  83, 79, 70, 84, 87, 65, 82, 69   <115>
1880 DATA  13, 13, 32, 32, 32, 32, 32, 32   <133>
1890 DATA  32, 32, 32, 77, 65, 68, 69, 32   <193>
1900 DATA 4172                              <199>
1910 :                                      <183>
1920 REM PROGRAMM-BLOCK # 7                 <44>
1930 DATA  66, 89, 32, 74, 79, 65, 67, 72   <96>
1940 DATA  73, 77, 32, 84, 72, 69, 69, 83   <149>
1950 DATA  13, 13, 13, 32, 32, 85, 83, 69   <181>
1960 DATA  58, 32, 62, 83, 65, 86, 69, 32   <14>
1970 DATA  32, 18, 34, 42, 46, 46, 46, 34   <86>
1980 DATA 146, 32, 79, 82, 32, 18, 34, 46   <91>
1990 DATA  46, 46, 46, 34, 13, 32, 32, 32   <90>
2000 DATA  32, 32, 32, 32, 62, 86, 69, 82   <11>
2010 DATA  73, 70, 89, 13, 32, 32, 32, 32   <124>
2020 DATA  32, 32, 32, 62, 76, 79, 65, 68   <189>
2030 DATA 4224                              <66>
2040 :                                      <57>
2050 REM PROGRAMM-BLOCK # 8                 <192>
2060 DATA  32, 40, 61, 32, 76, 79, 65, 68   <205>
2070 DATA  41, 13, 13,  0,162, 72,154,169   <161>
2080 DATA   0,133,214, 32,115,  4,240, 11   <224>
2090 DATA  32, 72,156,201, 18,144,  2,169   <200>
2100 DATA  17,133,214,160,  0,196,214,176   <223>
2110 DATA   9, 32,176,  4,153, 35, 16,200   <191>
2120 DATA 208,243,192, 17,176,  8,169, 32   <33>
2130 DATA 153, 35, 16,200,208,244, 32, 79   <135>
2140 DATA 255, 84, 85, 82, 66, 79, 45, 83   <54>
2150 DATA  65, 86, 73, 78, 71, 13,  0, 32   <168>
2160 DATA 7564                              <228>
2170 :                                      <188>
2180 REM PROGRAMM-BLOCK # 9                 <83>
2190 DATA  69, 20,162, 80,160,  1,134,178   <0>
2200 DATA 132,179,162, 84,160,  2,134,157   <69>
2210 DATA 132,158,169,  0,133,154,162, 35   <26>
2220 DATA 160, 16,134,175,132,176,169,179   <232>
2230 DATA 133,171, 32, 52,242,144,  5,162   <28>
2240 DATA  30, 76,131,134, 32,100,227,165   <186>
2250 DATA   1, 41,245,133,  1,162,  0,134   <230>
2260 DATA 212,134,213, 56,  8, 40,181, 43   <3>
2270 DATA 149,208,245, 45,149,210,  8,232   <82>
2280 DATA 224,  2,208,241, 40,169,208,141   <57>
2290 DATA 9817                              <110>
2300 :                                      <62>
2310 REM PROGRAMM-BLOCK # 10                <174>
2320 DATA 223,  7,160,  8,132,215, 32,148   <97>
2330 DATA  20,198,214,208,249,198,215,208   <180>
2340 DATA 245, 32,133, 20, 32,128, 20, 32   <73>
2350 DATA 217,  7, 24, 72,101,212,133,212   <175>
2360 DATA 144,  2,230,213,104, 32,103, 20   <10>
2370 DATA 230,208,208,  2,230,209,230,210   <12>
2380 DATA 208,226,230,211,208,222, 32,143   <114>
2390 DATA  20,165,212, 32,103, 20,165,213   <154>
2400 DATA  32,103, 20,160,  8,132,215, 32   <204>
2410 DATA 143, 20,198,214,208,249,198,215   <32>
2420 DATA 10987                             <56>
2430 :                                      <193>
2440 REM PROGRAMM-BLOCK # 11                <66>
2450 DATA 208,245,165,  1,  9,  8,133,  1   <113>
2460 DATA  56,110,252,  7, 32,120,227, 76   <255>
2470 DATA   3,128,162, 72,154, 32, 69, 20   <215>
2480 DATA 169,  0,133,171,169,  1, 32, 74   <229>
2490 DATA 240,  8,162,  3,189,182, 19,221   <114>
2500 DATA  51,  3,208,  6,202, 16,245, 40   <91>
2510 DATA 144, 24, 32, 79,255, 13, 63, 86   <227>
2520 DATA  79, 82, 83, 80, 65, 78, 78, 58   <37>
2530 DATA   0,162, 28, 76,131,134, 80,  1   <124>
2540 DATA  84,  2, 32,100,227, 32,141,227   <182>
2550 DATA 7590                              <99>
2560 :                                      <68>
2570 REM PROGRAMM-BLOCK # 12                <215>
2580 DATA 162,  0,134,212,134,213,202,134   <246>
2590 DATA   2,166, 43,164, 44,134,208,132   <201>
2600 DATA 209, 32, 56,  2, 32, 56,  2,176   <146>
2610 DATA   3, 32, 31,  2, 32, 56,  2,176   <72>
2620 DATA 251, 32, 53,  2, 32,  8,  2,160   <22>
2630 DATA   0,209,208,240,  2,  6,  2, 24   <21>
2640 DATA 101,212,133,212,144,  2,230,213   <216>
2650 DATA 230,208,208,  2,230,209, 32, 53   <158>
2660 DATA   2,144,225, 32,  8,  2, 69,212   <146>
2670 DATA 240,  2,  6,  2, 32,  8,  2, 69   <73>
2680 DATA 7688                              <13>
2690 :                                      <198>
2700 REM PROGRAMM-BLOCK # 13                <107>
2710 DATA 213,240,  2,  6,  2, 32,  8,  2   <158>
2720 DATA 162,  0,200,208,253,232,208,250   <60>
2730 DATA 165,  1,  9,  8,133,  1, 56,110   <161>
2740 DATA 252,  7, 32,120,227,230,  2,208   <66>
2750 DATA  17, 32, 79,255, 13, 70, 73, 76   <173>
2760 DATA  69, 32, 79, 75, 46, 13,  0, 76   <206>
2770 DATA   3,128, 76,177, 19,162, 80,169   <158>
2780 DATA   2,157,  0,  1,232,208,250,160   <76>
2790 DATA   0,185,213, 16,153,  0,  2,200   <124>
2800 DATA 192, 83,208,245,169,192,133,154   <18>
2810 DATA 8553                              <105>
2820 :                                      <73>
2830 REM PROGRAMM-BLOCK # 14                <0>
2840 DATA 169,  1,133,174,133,173, 96,162   <73>
2850 DATA   8,134,216, 74, 72, 32,126, 20   <202>
2860 DATA 173, 25,255, 73,127,141, 25,255   <234>
2870 DATA 104,198,216,208,238, 96,176, 15   <251>
2880 DATA 162, 74, 32,155, 20,162, 56, 32   <94>
2890 DATA 155, 20,162, 94, 76,155, 20,162   <53>
2900 DATA  74, 32,155, 20,162, 94, 32,155   <216>
2910 DATA  20,162, 56,160,  0,169, 16,141   <126>
2920 DATA   9,255, 44,  9,255,240,251,142   <169>
2930 DATA   2,255,140,  3,255,165,  1, 41   <5>
2940 DATA 9100                              <191>
2950 :                                      <203>
2960 REM PROGRAMM-BLOCK # 15                <148>
2970 DATA 253,133,  1,169, 16,141,  9,255   <100>
2980 DATA  44,  9,255,240,251,142,  2,255   <185>
2990 DATA 140,  3,255,165,  1,  9,  2,133   <124>
3000 DATA   1, 96,  0,  0,  0,  0,  0,  0   <246>
3010 DATA 2980                              <47>
  ENDE DES LISTINGS
```

RIESEN-LABYRINTH

Schnell eingetippt – toller Spaß

Bei diesem Spiel sind Sie in ein Labyrinth eingeschlossen und müssen innerhalb von 30 Minuten den Ausgang finden.
Diese Zeitvorgabe sieht sehr groß aus, ist aber gerechtfertigt. Der Spieler ist in einem Raum des Labyrinths und kann durch Türen in den Nächsten gelangen. Der Überblick ist also stark eingeschränkt. Eine Aufsicht von oben, um sich zu orientieren, ist nicht möglich. Es empfiehlt sich daher, während des Spielens einen Lageplan anzulegen, um den Überblick nicht völlig zu verlieren. Sie können sich freuen, wenn Sie mit Hilfe dieser "Landkarte" beim dritten Anlauf zum das Ziel gefunden haben. Die Bewegung von einem Raum zum anderen erfolgt durch die Tasten N-orden, O-sten, S-üden und W-esten. Bereits betretene Räume werden markiert, der Rahmen erscheint dann Rot, im Gegensatz zu Blau bei "Neuentdeckungen".
Wollen Sie das Spiel vorzeitig abbrechen, so haben Sie die Möglichkeit, den aktuellen Spielstand auf Kassette abzuspeichern, um ihn bei einem späteren Spiel wieder zu laden. Die dadurch vergangene Spielzeit wird jeweils berücksichtigt, da die Datasette ja nicht gerade der schnellste Massenspeicher ist.
Die Spielfläche besteht aus einem Quadrat von 20 x 20 Räumen. Die Aufgabe besteht darin, vom Raum 0/0 zum Raum 19/19 zu gelangen, also im Endeffekt die Fläche diagonal in Nord-Ost-Richtung zu überqueren. In der Praxis wird der Weg, wie zu vermuten ist, kreuz und quer verlaufen.
Die Zu- und Ausgänge der 400 Räume sind im Programm in DATA-Zeilen gespeichert. Wie man sich denken kann, führt ein Fehler beim Abtippen dieser Zeilen zu einer erheblichen Veränderung des Labyrinths, das dann vielleicht gar nicht mehr lösbar ist. Aus diesem Grunde werden die DATA-Zeilen per Prüfsumme gecheckt, bevor das Programm startet. Die kleine Verzögerung beim Einlesen sichert die Richtigkeit des ganzen Spiels.



C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
10 REM ******************                <85>
20 REM *                *                <117>
30 REM * RIESEN-LABYRINTH *              <19>
40 REM *                *                <137>
50 REM * DANIEL SCHIEMANN *             <249>
60 REM *     NEUSTADT   *               <134>
70 REM *                *                <167>
80 REM ******************                <155>
90 REM                                   <233>
100 REM INITIALISIERUNG                  <20>
110 DIML$(19,19),L%(19,19)               <251>
120 COLOR0,8:COLOR4,6:X=0:Y=0:T$="000000" <24>
130 GOSUB1040                            <193>
140 CHAR1,25,12,"{RVSON SPACE}L{SPACE RVSOFF SPA
CE}={SPACE}ALTES"                        <233>
150 CHAR1,25,13,"SPIEL{SPACE}LADEN"      <164>
160 CHAR1,25,15,"{RVSON SPACE}+{SPACE RVSOFF SPA
CE}={SPACE}START"                        <4>
170 GETKEYA$                             <144>
180 IFA$="L"THENGOSUB1280                <162>
190 IFA$="+"THEN210                      <141>
200 GOTO170                              <24>
210 CHAR1,25,20,"BITTE{SPACE}WARTEN"     <194>
220 FORI=0TO19                           <174>
230 FORJ=0TO19                           <186>
240 READL$(J,I)                          <220>
250 FORK=1TO4                            <87>
260 S=S+ASC(MID$(L$(I,J),K,1))           <223>
```

```
270 NEXTK                                                    <39>
280 NEXTJ:CHAR1,25,22,STR$(19-I)+"{SPACE}"                   <17>
290 NEXTI                                                    <55>
300 TI$=T$                                                   <195>
310 IFS<>54011THENPRINT"{CLEAR FLASHON}DATA-FEHL
ER":STOP                                                     <213>
320 REM HAUPTSCHLEIFE                                        <147>
330 GOSUB1040:Z$=""                                          <136>
340 IFL%(X,Y)=1THENCOLOR4,3:ELSECOLOR4,7                     <6>
350 CHAR1,25,12,"GEHEN{SPACE}SIE{SPACE}NACH"                 <101>
360 FORI=1TO4                                                <194>
370 IFMID$(L$(X,Y),I,1)<>"."THENRI(I)=1:ELSERI(I
)=0                                                          <231>
380 NEXTI                                                    <145>
390 IFRI(1)=1THENCHAR1,27,14,"{RVSON}N{RVSOFF}OR
DEN":GOSUB670                                                <251>
400 IFRI(2)=1THENCHAR1,27,16,"{RVSON}O{RVSOFF}ST
EN":GOSUB760                                                 <15>
410 IFRI(3)=1THENCHAR1,27,18,"{RVSON}S{RVSOFF}UE
DEN":GOSUB890                                                <150>
420 IFRI(4)=1THENCHAR1,27,20,"{RVSON}W{RVSOFF}ES
TEN":GOSUB910                                                <16>
430 Z$="":FORI=1TO5STEP2                                     <201>
440 Z$=Z$+":"+MID$(TI$,I,2)                                  <164>
450 NEXTI                                                    <215>
460 CHAR1,25,22,"ZEIT"+Z$                                    <193>
470 CHAR1,28,24,"{RVSON}ESC{RVSOFF SPACE}={SPACE
}ENDE{UP}"                                                   <17>
480 GETA$                                                    <112>
490 REM SPIELZEIT 30 MINUTEN                                 <62>
500 IF VAL(TI$)>003000THEN1750                               <153>
510 I=5                                                      <75>
520 FORJ=1TO4                                                <101>
530 IFA$=MID$(L$(X,Y),J,1)THENI=J                            <24>
540 NEXTJ                                                    <52>
550 IFI<5THEN580                                             <187>
560 IFA$=CHR$(27)THEN1410                                    <251>
570 GOTO430                                                  <134>
580 L%(X,Y)=1                                                <14>
590 ONIGOTO600,610,620,630                                   <192>
600 Y=Y+1:GOTO640                                            <62>
610 X=X+1:GOTO640                                            <68>
620 Y=Y-1:GOTO640                                            <86>
630 X=X-1                                                    <221>
640 I=0                                                      <191>
650 IFX=19ANDY=19THEN1650                                    <183>
660 GOTO330                                                  <222>
670 REM TUER VORNE (NORDEN)                                  <236>
680 CHAR1,10,11,"{CA SC2 CS}"                                <8>
690 CHAR1,10,12,"{S- SPACE2 S-}"                             <176>
700 CHAR1,10,13,"{S- SPACE2 S-}"                             <194>
710 CHAR1,10,14,"{S- SPACE2 S-}"                             <212>
720 CHAR1,10,15,"{S- SPACE2 S-}"                             <230>
730 CHAR1,10,16,"{S- SPACE2 S-}"                             <248>
740 CHAR1,10,17,"{CY SPACE2 CY}"                             <84>
750 RETURN                                                   <126>
760 REM TUER RECHTS (OSTEN)                                  <223>
770 CHAR1,19,11,"{SO2 SP}"                                   <120>
780 CHAR1,19,12,"{CG CH CM}"                                 <99>
790 CHAR1,19,13,"{CG2 CM}"                                   <193>
800 CHAR1,19,14,"{CG2 CM}"                                   <211>
810 CHAR1,19,15,"{CG2 CM}"                                   <229>
820 CHAR1,19,16,"{CG2 CM}"                                   <247>
830 CHAR1,19,17,"{CG2 CM}"                                   <9>
840 CHAR1,19,18,"{SL CG CM}"                                 <200>
850 CHAR1,19,19,"{SPACE2 CM}"                                <58>
860 CHAR1,19,20,"{SPACE2 CM}"                                <3>
870 CHAR1,19,21,"{SPACE2 CM}"                                <21>
880 RETURN                                                   <1>
890 REM TUER SUEDEN NICHT SICHTBAR                           <18>
900 RETURN                                                   <21>
910 REM TUER LINKS (WESTEN)                                  <176>
920 CHAR1,02,11,"{SO SP2}"                                   <243>
930 CHAR1,02,12,"{CG CN CM}"                                 <90>
940 CHAR1,02,13,"{CG CN CM}"                                 <108>
950 CHAR1,02,14,"{CG CN CM}"                                 <126>
960 CHAR1,02,15,"{CG CN2}"                                   <183>
970 CHAR1,02,16,"{CG CN CM}"                                 <162>
980 CHAR1,02,17,"{CG CN CM}"                                 <180>
990 CHAR1,02,18,"{CG CN S@}"                                 <189>
1000 CHAR1,02,19,"{CH SPACE2}"                               <42>
1010 CHAR1,02,20,"{CH SPACE2}"                               <243>
1020 CHAR1,02,21,"{CH SPACE2}"                               <5>
1030 RETURN                                                  <152>
1040 REM BILD                                                <134>
1050 SCNCLR                                                  <6>
1060 FORI=0TO6                                               <137>
1070 CHAR1,I,I,"{SM}"                                        <123>
1080 CHAR1,23-I,I,"{SN}"                                     <75>
1090 CHAR1,I,23-I,"{SN}"                                     <22>
1100 CHAR1,23-I,23-I,"{SM}"                                  <254>
1110 NEXTI                                                   <110>
1120 FORI=7TO16                                              <61>
1130 CHAR1,I,6,"{CP}"                                        <55>
1140 CHAR1,I,17,"{CY}"                                       <19>
1150 CHAR1,6,I,"{CN}"                                        <68>
1160 CHAR1,17,I,"{CH}"                                       <64>
1170 NEXTI                                                   <170>
1180 FORI=0TO24                                              <85>
1190 CHAR1,24,I,"{CH}"                                       <83>
1200 NEXTI                                                   <200>
1210 CHAR1,28,1,"{RVSON}RIESEN-{RVSOFF}"                     <184>
1220 CHAR1,27,3,"{RVSON}LABYRINTH{RVSOFF}"                   <162>
1230 CHAR1,25,6,"KOORDINATEN"                                <199>
1240 CHAR1,27,7,"DIESES{SPACE}RAUMS"                         <139>
1250 CHAR1,26,9,"X=":PRINTX                                  <202>
1260 CHAR1,33,9,"Y=":PRINTY                                  <226>
1270 RETURN                                                  <136>
1280 REM SPIEL LADEN                                         <251>
1290 SCNCLR                                                  <247>
1300 OPEN1,1,0,"LABYRINTH"                                   <0>
1310 INPUT#1,T$                                              <145>
1320 INPUT#1,X                                               <247>
1330 INPUT#1,Y                                               <5>
1340 FORI=0TO19                                              <19>
1350 FORJ=0TO19                                              <31>
1360 INPUT#1,L%(I,J)                                         <43>
1370 NEXTJ                                                   <117>
1380 NEXTI                                                   <125>
1390 CLOSE1                                                  <117>
1400 RETURN                                                  <11>
1410 REM SPIEL BEENDEN                                       <126>
1420 SCNCLR                                                  <121>
1430 CHAR1,10,5,"SPIELSTAND{SPACE}SPEICHERN"                 <1>
1440 CHAR1,18,7,"{FLASHON}J{SPACE}/{SPACE}N{FLAS
HOFF}"                                                       <91>
1450 GETKEYA$                                                <149>
1460 IFA$="J"THENGOSUB1520                                   <116>
1470 CHAR1,12,10,"SPIEL{SPACE}BEENDEN"                       <109>
1480 CHAR1,18,12,"{FLASHON}J{SPACE}/{SPACE}N{FLA
SHOFF}"                                                      <102>
1490 GETKEYA$                                                <189>
1500 IFA$="N"THEN330                                         <159>
1510 END                                                     <107>
1520 REM SPIEL SPEICHERN                                     <59>
1530 SCNCLR                                                  <231>
1540 OPEN1,1,1,"LABYRINTH"                                   <247>
1550 PRINT#1,T$                                              <150>
1560 PRINT#1,X                                               <252>
1570 PRINT#1,Y                                               <10>
1580 FORI=0TO19                                              <4>
1590 FORJ=0TO19                                              <16>
1600 PRINT#1,L%(I,J)                                         <48>
1610 NEXTJ                                                   <102>
1620 NEXYI                                                   <11>
1630 CLOSE1                                                  <102>
1640 RETURN                                                  <252>
1650 REM AM ZIEL                                             <247>
1660 FORI=1TO16                                              <67>
1670 FORJ=0TO7                                               <245>
1680 COLOR4,I,J                                              <37>
1690 NEXTJ                                                   <182>
1700 NEXTI                                                   <190>
1710 SCNCLR                                                  <156>
1720 CHAR1,5,5,"{FLASHON}SIE{SPACE}SIND{SPACE}AM
{SPACE}ZIEL{FLASHOFF}"                                       <158>
1730 CHAR1,10,10,"IHRE{SPACE}ZEIT"+Z$                        <121>
1740 END                                                     <82>
1750 REM ZEIT VORBEI                                         <154>
1760 SCNCLR                                                  <206>
1770 CHAR1,5,5,"VERLOREN,{SPACE}ZEIT{SPACE}UEBER
SCHRITTEN"                                                   <103>
1780 END                                                     <122>
1790 REM LABYRINTH 20 X 20 IN DATAS                          <156>
1800 DATA NO..,N..W,NO..,.O.W,.O.W,NO.W,.O.W,NO.
W,.O.W,NO.W                                                  <216>
1810 DATA NO.W,.O.W,N..W,NO..,NO.W,.O.W,N..W,NO.
.,...W,N...                                                  <119>
1820 DATA N.S.,.OS.,NOSW,.O.W,N..W,.OS.,...W,N.S
.,NO..,..SW                                                  <85>
1830 DATA ..S.,N...,NOS.,..SW,NOS.,...W,N.S.,.OS
.,.O.W,N.S.                                                  <135>
```

```
1840 DATA .OS.,N..W,..S.,NO..,N.SW,NO..,...W,NOS
.,.OSW,NO.W                                      <226>
1850 DATA .O.W,N.SW,..S.,N...,N.S.,N...,.OS.,.O.
W,.O.W,N.SW                                      <193>
1860 DATA N...,.OS.,N..W,N.S.,.OS.,.OSW,N..W,..S
.,N...,.OS.                                      <14>
1870 DATA N..W,NOS.,...W,NOS.,.OSW,..SW,NO..,.O.
W,N..W,N.S.                                      <50>
1880 DATA NOS.,...W,NOS.,.OSW,N..W,N...,.OS.,.O.
W,.OSW,...W                                      <251>
1890 DATA NOS.,.OSW,...W,.OS.,N..W,NO..,N.SW,N..
.,N.S.,N.S.                                      <21>
1900 DATA .OS.,NO.W,NOSW,...W,.OS.,NOSW,NO.W,.O.
W,.O.W,N..W                                      <139>
1910 DATA .OS.,NO.W,...W,.O..,N.SW,NOS.,..SW,NOS
.,..SW,N.S.                                      <228>
1920 DATA NO..,..SW,NOS.,NO.W,NO.W,..SW,..S.,NO.
.,.O.W,..SW                                      <131>
1930 DATA N...,.OS.,N..W,.O..,.OSW,N.SW,.O..,NOS
W,...W,N.S.                                      <197>
1940 DATA N.S.,NO..,..SW,N.S.,.OS.,NO.W,N..W,..S
.,.O..,NO.W                                      <82>
1950 DATA .OSW,N..W,.OS.,N..W,.O..,..SW,NO..,.OS
W,...W,N.S.                                      <215>
1960 DATA NOS.,.OSW,NO.W,.OSW,NO.W,..SW,N.S.,NO.
.,.O.W,NOSW                                      <87>
1970 DATA ...W,.OS.,N..W,.OS.,.O.W,...W,.OS.,.O.
W,N..W,N.S.                                      <23>
1980 DATA .OS.,N..W,.OS.,N..W,.OS.,.O.W,..SW,N.S
.,N...,N.S.                                      <172>
1990 DATA .O..,.O.W,.OSW,.O.W,N..W,NO..,.O.W,N..
W,N.S.,N.S.                                      <63>
2000 DATA NO..,NOSW,...W,N.S.,NO..,NO.W,.O.W,..S
W,N.S.,.OS.                                      <115>
2010 DATA .O.W,.O.W,.O.W,...W,.OS.,N.SW,N...,.OS
.,..SW,N.S.                                      <115>
2020 DATA ..S.,.OS.,N..W,..S.,N.S.,..S.,N...,.O.
.,.OSW,.O.W                                      <178>
2030 DATA .O.W,NO.W,.O.W,NO.W,N..W,..S.,NOS.,.O.
W,.O.W,..SW                                      <62>
2040 DATA N...,NO..,..SW,.O..,NOSW,.O.W,.OSW,...
W,NO..,...W                                      <22>
2050 DATA .O..,N.SW,NO..,..SW,NOS.,.O.W,..SW,N..
.,NO..,...W                                      <122>
2060 DATA NOS.,.OSW,N..W,NO..,..SW,NO..,NO.W,.O.
W,..SW,NO..                                      <110>
2070 DATA N..W,..S.,NO..,N..W,..S.,NO..,N..W,.OS
.,.OSW,N..W                                      <42>
2080 DATA N.S.,N...,N.S.,.OS.,NO.W,..SW,..S.,NO.
.,.O.W,N.SW                                      <236>
2090 DATA NOS.,...W,N.S.,.OS.,...W,N.S.,.OS.,N..
W,NO..,N.SW                                      <188>
2100 DATA .OS.,N.SW,NOS.,N..W,N.S.,.O..,NO.W,.OS
W,...W,N.S.                                      <231>
2110 DATA .OS.,N..W,N.S.,NO..,N..W,.OS.,.O.W,NOS
W,N.SW,N.S.                                      <11>
2120 DATA N...,N.S.,N.S.,N.S.,.OS.,N..W,N.S.,.O.
.,N..W,..S.                                      <180>
2130 DATA N...,N.S.,..S.,N.S.,.OS.,.O.W,NO.W,..S
W,..S.,N.S.                                      <88>
2140 DATA NOS.,N.SW,..S.,.OS.,N..W,.OS.,..SW,NO.
.,NOSW,...W                                      <11>
2150 DATA N.S.,NOS.,.O.W,.OSW,NO.W,N..W,N.S.,NO.
.,N..W,..S.                                      <75>
2160 DATA N.S.,..S.,NO..,N..W,..S.,NO..,.O.W,N.S
W,NOS.,N..W                                      <185>
2170 DATA N.S.,N.S.,.O..,.O.W,N.SW,N.S.,N.S.,N.S
.,.OS.,N..W                                      <17>
2180 DATA .OS,.O.W,..SW,.OS.,.O.W,.OSW,...W,..S
.,..S.,.OS.                                      <176>
2190 DATA ..SW,.OS.,.O.W,...W,..S.,..S.,.OS.,.OS
W,...W,..S.                                      <62>
  ENDE DES LISTINGS
```

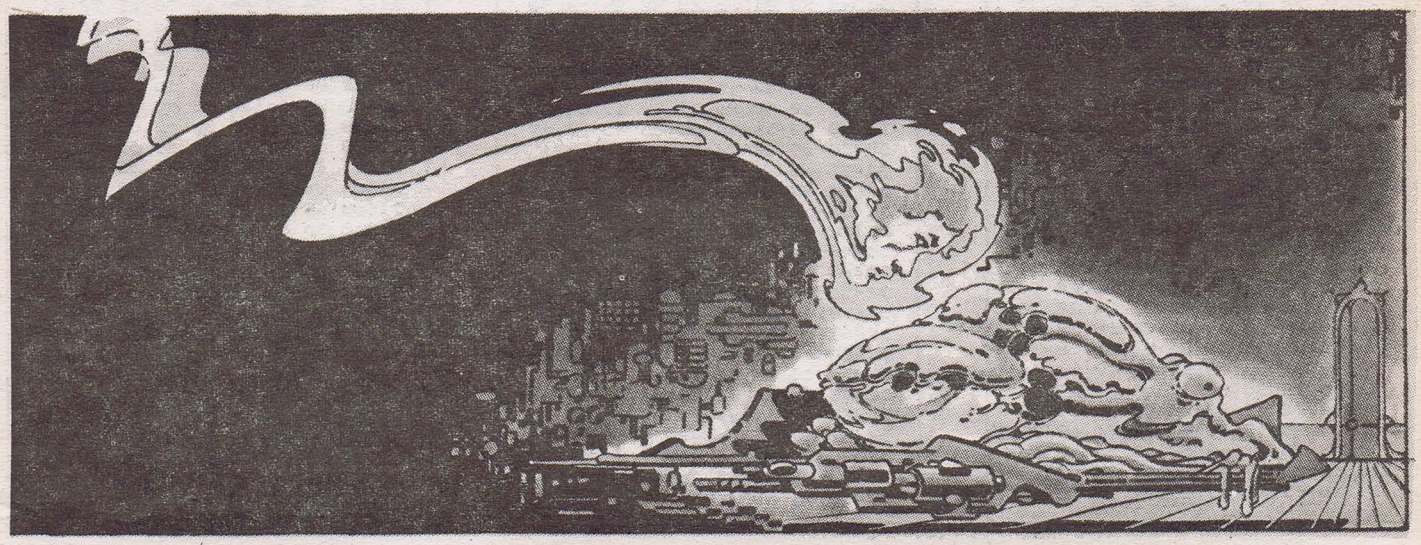
Der Spielhallenhit

CRAZY WORMS

Aus der Spielhalle werden Sie dieses Spiel sicher kennen. Widerlich aufdringliche Gliederwürmer kommen aus dem All und bewegen sich schlängelnd auf die Erde zu. Mit einer Laserkanone sollen Sie nun die Erde vor diesen Tieren bewahren. Sie steuern wahlweise mit den Tasten "S" (hoch), "X" (runter), ":" (links), ":" (rechts), SPACE (Feuer) oder mit dem Joystick in Port 1.

Auf dem Blidschirm befinden sich noch eine Reihe von Pilzen, die ebenfalls abgeschossen werden können. Nach dem vierten Treffer verschwindet ein Pilz vollständig.

Punktvergabe:
Wurmteil: 100 Punkte
Piztreffer: 1 Punkt



Eingabehinweise
Nach Einschalten Ihres Computers geben SIe MONITOR (RETURN) ein. Danach tippen Sie das Listing ein, wie es abgedruckt ist.
Nachdem Sie das Programm vollständig abgetippt haben, speichern Sie es bitte sofort ab.
1. auf Diskette **S"CRAZY WORMS",0,1200,2DBA**

2. auf Kassette **S"CRAZY WORMS",1,1200,2ABA**

Nun verlassen Sie den Monitor mit **X** und können das Spiel mit **SYS 4608** starten.

Fehlerkorrektur
Läuft das Programm nicht, dann müssen Sie es neu einladen und anschließend in den MONITOR gehen und mit **M 1200** mit der Fehlersuche beginnen. Haben Sie Fehler gefunden und verbessert, vergessen Sie nicht das Programm erneut abzuspeichern.

Einladen
Von Diskette LOAD "CRAZY WORMS",8,1
von Kassette MONITOR (RETURN), L (RETURN)
Gestartet wird mit **SYS 4608**
Viel Spaß!

```
monitor

   pc  sr ac xr yr sp
; ffff 00 ff ff ff f8

>1200 a9 00 8d f8 07 85 d0 85
>1208 d2 a9 d0 85 d1 a9 3c 85
>1210 d3 8d 13 ff a0 00 b1 d0
>1218 91 d2 c8 d0 f9 e6 d1 e6
>1220 d3 a5 d3 c9 40 d0 ed a9
>1228 c0 8d 12 ff a9 3b 85 34
>1230 85 38 a9 f6 85 33 85 37
>1238 4c 7c 12 02 3c 7e ff ff
>1240 ff 18 18 18 3c 7e ff ff
>1248 ff 08 08 00 3c 7e ff e7
>1250 e3 00 00 00 3c 7e cf 08
>1258 00 00 00 00 00 00 00 00
>1260 00 00 00 00 18 18 18 18
>1268 3c 7e ff ff 00 00 18 18
>1270 18 00 00 00 42 3c 7e 7e
>1278 7e 7e 3c 42 a2 00 bd 3c
>1280 12 9d c0 3f e8 e0 40 d0
>1288 f5 4c 86 2a 00 3a 10 01
>1290 00 99 22 1e 93 22 3a e7
>1298 30 2c 31 3a e7 34 2c 31
>12a0 3a 81 54 b2 31 a4 35 30
>12a8 3a 58 b2 b5 28 bb 28 31
>12b0 29 ac 37 36 30 29 aa 38
>12b8 30 3a 97 33 30 37 32 aa
>12c0 58 2c 31 32 30 00 4a 10
>12c8 02 00 97 32 30 34 38 aa
>12d0 58 2c 36 3a 82 00 7e 10
>12d8 03 00 99 22 05 13 20 20
>12e0 53 43 4f 52 45 3a 30 30
>12e8 30 30 30 30 20 20 20 20
>12f0 20 20 20 20 20 20 20 20
>12f8 20 48 49 3a 30 30 30 30
>1300 30 30 22 3a 9e 34 39 32
>1308 37 00 00 00 a2 00 bd 8c
>1310 12 9d 00 10 e8 e0 80 d0
>1318 f5 a9 90 20 d2 ff a9 00
>1320 8d 15 ff 8d 19 ff a9 04
>1328 85 ef a9 52 a2 55 a0 4e
>1330 8d 27 05 8e 28 05 8c 29
>1338 05 a9 0d 8d 2a 05 60 20
```

```
>1340 e0 2a 4c f0 1f 00 02 01
>1348 02 02 02 03 02 04 02 05
>1350 02 06 02 07 02 08 02 09
>1358 02 0a 02 0b 02 00 0c 28
>1360 0c 50 0c 78 0c a0 0c c8
>1368 0c f0 0c 18 0d 40 0d 68
>1370 0d 90 0d b8 0d e0 0d 08
>1378 0e 30 0e 58 0e 80 0e a8
>1380 0e d0 0e f8 0e 20 0f 48
>1388 0f 70 0f 98 0f c0 0f 00
>1390 00 00 00 00 00 ea a2 fe
>1398 a0 00 e8 e8 c8 cc 00 3b
>13a0 d0 f8 bd 03 3b 8d 4f 3b
>13a8 e8 bd 03 3b 8d 50 3b ad
>13b0 01 3b c9 01 f0 03 4c b4
>13b8 14 ee 4f 3b ad 4f 3b c9
>13c0 28 d0 5d a9 27 8d 4f 3b
>13c8 ad 02 3b c9 01 d0 2a ee
>13d0 50 3b ad 50 3b c9 19 d0
>13d8 18 a9 18 8d 50 3b a9 00
>13e0 8d 01 3b 8d 02 3b ce 50
>13e8 3b ea ea ea ea ea 4c ae
>13f0 15 a9 00 8d 01 3b 4c ce
>13f8 1a ce 50 3b ad 50 3b c9
>1400 01 d0 15 a9 02 8d 50 3b
>1408 a9 00 8d 01 3b a9 01 8d
>1410 02 3b ee 50 3b 4c ae 15
>1418 a9 00 8d 01 3b 4c ae 15
>1420 ad 50 3b 2a aa bd 1b 3b
>1428 85 d8 e8 bd 1b 3b 85 d9
>1430 ad 4f 3b c9 00 f0 12 e9
>1438 01 8d 4e 3b a5 d8 6d 4e
>1440 3b c5 d8 b0 02 e6 d9 85
>1448 d8 a0 00 b1 d8 c9 7d d0
>1450 03 20 3e 2a c9 20 d0 03
>1458 4c ae 15 ad 02 3b c9 01
>1460 d0 28 ce 4f 3b ee 50 3b
>1468 ad 50 3b c9 19 d0 13 a9
>1470 18 8d 50 3b a9 00 8d 01
>1478 3b 8d 02 3b ce 50 3b 4c
>1480 ae 15 a9 00 8d 01 3b 4c
>1488 ae 15 ce 4f 3b ce 50 3b
>1490 ad 50 3b c9 01 d0 15 a9
>1498 02 8d 50 3b a9 00 8d 01
>14a0 3b a9 01 8d 02 3b ee 50
```

```
>14a8 3b 4c ae 15 a9 00 8d 01
>14b0 3b 4c ae 15 ce 4f 3b ad
>14b8 4f 3b c9 ff d0 5d a9 00
>14c0 8d 4f 3b ad 02 3b c9 01
>14c8 d0 2a ee 50 3b ad 50 3b
>14d0 c9 19 d0 18 a9 18 8d 50
>14d8 3b a9 01 8d 01 3b a9 00
>14e0 8d 02 3b ce 50 3b ea ea
>14e8 ea 4c ae 15 a9 01 8d 01
>14f0 3b 4c ce 1a ce 50 3b ad
>14f8 50 3b c9 01 d0 15 a9 02
>1500 8d 50 3b a9 01 8d 01 3b
>1508 a9 01 8d 02 3b ee 50 3b
>1510 4c ae 15 a9 01 8d 01 3b
>1518 4c ae 15 ad 50 3b 2a aa
>1520 bd 1b 3b 85 d8 e8 bd 1b
>1528 3b 85 d9 ad 4f 3b c9 00
>1530 f0 78 e9 01 8d 4e 3b a5
>1538 d8 6d 4e 3b c5 d8 b0 02
>1540 e6 d9 85 d8 a0 00 b1 d8
>1548 c9 7d d0 03 20 3e 2a c9
>1550 20 d0 03 4c ae 15 ad 02
>1558 3b c9 01 d0 2a ee 4f 3b
>1560 ee 50 3b ad 50 3b c9 19
>1568 d0 15 a9 18 8d 50 3b a9
>1570 01 8d 01 3b a9 00 8d 02
>1578 3b ce 50 3b 4c ae 15 a9
>1580 01 8d 01 3b 4c ae 15 ee
>1588 4f 3b ce 50 3b ad 50 3b
>1590 c9 01 d0 15 a9 02 8d 50
>1598 3b a9 01 8d 01 3b a9 01
>15a0 8d 02 3b ee 50 3b 4c ae
>15a8 15 a9 01 8d 01 3b a2 00
>15b0 ac 04 3b e8 e8 88 d0 fb
>15b8 bd 1b 3b 85 d8 bd 1c 3b
>15c0 85 d9 ac 03 3b c0 00 f0
>15c8 0d e6 d8 a5 d8 c9 00 d0
>15d0 02 e6 d9 88 d0 f3 a9 20
>15d8 a0 00 91 d8 a2 00 bd 05
>15e0 3b 9d 51 3b e8 e0 16 d0
>15e8 f5 a2 00 bd 51 3b 9d 03
>15f0 3b e8 e0 16 d0 f5 a2 fe
>15f8 ac 00 3b e8 e8 88 d0 fb
>1600 ad 4f 3b 9d 03 3b ad 50
>1608 3b 9d 04 3b a9 00 8d 4d
>1610 3b a2 00 ac 04 3b e8 e8
>1618 88 d0 fb bd 1b 3b 85 d8
>1620 bd 1c 3b 85 d9 ac 03 3b
>1628 c0 00 f0 0d e6 d8 a5 d8
>1630 c9 00 d0 02 e6 d9 88 d0
>1638 f3 a5 d8 85 da a5 d9 e9
>1640 04 85 db a0 00 a9 7f 91
>1648 d8 91 da ee 4d 3b ad 4d
>1650 3b cd 00 3b d0 03 4c 63
>1658 19 a2 00 ac 06 3b e8 e8
>1660 88 d0 fb bd 1b 3b 85 d8
>1668 bd 1c 3b 85 d9 ac 05 3b
>1670 c0 00 f0 0d e6 d8 a5 d8
>1678 c9 00 d0 02 e6 d9 88 d0
>1680 f3 a5 d8 85 da a5 d9 e9
>1688 04 85 db a0 00 a9 7f 91
>1690 d8 91 da ee 4d 3b ad 4d
>1698 3b cd 00 3b d0 03 4c 63
>16a0 19 a2 00 ac 08 3b e8 e8
>16a8 88 d0 fb bd 1b 3b 85 d8
>16b0 bd 1c 3b 85 d9 ac 07 3b
>16b8 c0 00 f0 0d e6 d8 a5 d8
>16c0 c9 00 d0 02 e6 d9 88 d0
>16c8 f3 a5 d8 85 da a5 d9 e9
>16d0 04 85 db a0 00 a9 7f 91
>16d8 d8 91 da ee 4d 3b ad 4d
>16e0 3b cd 00 3b d0 03 4c 63
>16e8 19 a2 00 ac 0a 3b e8 e8
>16f0 88 d0 fb bd 1b 3b 85 d8
>16f8 bd 1c 3b 85 d9 ac 09 3b
>1700 c0 00 f0 0d e6 d8 a5 d8
>1708 c9 00 d0 02 e6 d9 88 d0
>1710 f3 a5 d8 85 da a5 d9 e9
>1718 04 85 db a0 00 a9 7f 91
>1720 d8 91 da ee 4d 3b ad 4d
>1728 3b cd 00 3b d0 03 4c 63
>1730 19 a2 00 ac 0c 3b e8 e8
>1738 88 d0 fb bd 1b 3b 85 d8
>1740 bd 1c 3b 85 d9 ac 0b 3b
>1748 c0 00 f0 0d e6 d8 a5 d8
>1750 c9 00 d0 02 e6 d9 88 d0
```

```
>1758 f3 a5 d8 85 da a5 d9 e9
>1760 04 85 db a0 00 a9 7f 91
>1768 d8 91 da ee 4d 3b ad 4d
>1770 3b cd 00 3b d0 03 4c 63
>1778 19 a2 00 ac 0e 3b e8 e8
>1780 88 d0 fb bd 1b 3b 85 d8
>1788 bd 1c 3b 85 d9 ac 0d 3b
>1790 c0 00 f0 0d e6 d8 a5 d8
>1798 c9 00 d0 02 e6 d9 88 d0
>17a0 f3 a5 d8 85 da a5 d9 e9
>17a8 04 85 db a0 00 a9 7f 91
>17b0 d8 91 da ee 4d 3b ad 4d
>17b8 3b cd 00 3b d0 03 4c 63
>17c0 19 a2 00 ac 10 3b e8 e8
>17c8 88 d0 fb bd 1b 3b 85 d8
>17d0 bd 1c 3b 85 d9 ac 0f 3b
>17d8 c0 00 f0 0d e6 d8 a5 d8
>17e0 c9 00 d0 02 e6 d9 88 d0
>17e8 f3 a5 d8 85 da a5 d9 e9
>17f0 04 85 db a0 00 a9 7f 91
>17f8 d8 91 da ee 4d 3b ad 4d
>1800 3b cd 00 3b d0 03 4c 63
>1808 19 a2 00 ac 12 3b e8 e8
>1810 88 d0 fb bd 1b 3b 85 d8
>1818 bd 1c 3b 85 d9 ac 11 3b
>1820 c0 00 f0 0d e6 d8 a5 d8
>1828 c9 00 d0 02 e6 d9 88 d0
>1830 f3 a5 d8 85 da a5 d9 e9
>1838 04 85 db a0 00 a9 7f 91
>1840 d8 91 da ee 4d 3b ad 4d
>1848 3b cd 00 3b d0 03 4c 63
>1850 19 a2 00 ac 14 3b e8 e8
>1858 88 d0 fb bd 1b 3b 85 d8
>1860 bd 1c 3b 85 d9 ac 13 3b
>1868 c0 00 f0 0d e6 d8 a5 d8
>1870 c9 00 d0 02 e6 d9 88 d0
>1878 f3 a5 d8 85 da a5 d9 e9
>1880 04 85 db a0 00 a9 7f 91
>1888 d8 91 da ee 4d 3b ad 4d
>1890 3b cd 00 3b d0 03 4c 63
>1898 19 a2 00 ac 16 3b e8 e8
>18a0 88 d0 fb bd 1b 3b 85 d8
>18a8 bd 1c 3b 85 d9 ac 15 3b
>18b0 c0 00 f0 0d e6 d8 a5 d8
>18b8 c9 00 d0 02 e6 d9 88 d0
>18c0 f3 a5 d8 85 da a5 d9 e9
>18c8 04 85 db a0 00 a9 7f 91
>18d0 d8 91 da ee 4d 3b ad 4d
>18d8 3b cd 00 3b d0 03 4c 63
>18e0 19 a2 00 ac 18 3b e8 e8
>18e8 88 d0 fb bd 1b 3b 85 d8
>18f0 bd 1c 3b 85 d9 ac 17 3b
>18f8 c0 00 f0 0d e6 d8 a5 d8
>1900 c9 00 d0 02 e6 d9 88 d0
>1908 f3 a5 d8 85 da a5 d9 e9
>1910 04 85 db a0 00 a9 7f 91
>1918 d8 91 da ee 4d 3b ad 4d
>1920 3b cd 00 3b d0 03 4c 63
>1928 19 a2 00 ac 1a 3b e8 e8
>1930 88 d0 fb bd 1b 3b 85 d8
>1938 bd 1c 3b 85 d9 ac 19 3b
>1940 c0 00 f0 0d e6 d8 a5 d8
>1948 c9 00 d0 02 e6 d9 88 d0
>1950 f3 a5 d8 85 da a5 d9 e9
>1958 04 85 db a0 00 a9 7f 91
>1960 d8 91 da 4c 04 1b ea ea
>1968 ea ea ea ea ea ea a2 00
>1970 ac 68 3b e8 e8 88 d0 fb
>1978 bd 1b 3b 85 d8 bd 1c 3b
>1980 85 d9 ac 67 3b c0 00 f0
>1988 0d e6 d8 a5 d8 c9 00 d0
>1990 02 e6 d9 88 d0 f3 a0 00
>1998 a9 20 91 d8 a5 c6 ea ea
>19a0 ea ea ea ea c9 12 d0 03
>19a8 4c 87 1d c9 2d d0 03 4c
>19b0 87 1d c9 13 d0 03 4c b0
>19b8 1d c9 32 d0 03 4c b0 1d
>19c0 c9 10 d0 03 4c 36 1d c9
>19c8 0d d0 03 4c 36 1d c9 11
>19d0 d0 03 4c 61 1d c9 17 d0
>19d8 03 4c 61 1d c9 16 d0 03
>19e0 4c 35 1a c9 3c d0 03 4c
>19e8 35 1a 4c 50 1a ce 67 3b
>19f0 ad 67 3b c9 ff d0 05 a9
>19f8 00 8d 67 3b 4c 50 1a ee
>1a00 67 3b ad 67 3b c9 28 d0
```

```
>1a08 05 a9 27 8d 67 3b 4c 50
>1a10 1a ce 68 3b ad 68 3b c9
>1a18 14 d0 05 a9 15 8d 68 3b
>1a20 4c 98 1a ee 68 3b ad 68
>1a28 3b c9 19 d0 05 a9 18 8d
>1a30 68 3b 4c 98 1a ad 6b 3b
>1a38 c9 01 f0 14 ad 67 3b 8d
>1a40 69 3b ad 68 3b 8d 6a 3b
>1a48 a9 01 8d 6b 3b ea ea ea
>1a50 a2 00 ac 68 3b e8 e8 88
>1a58 d0 fb bd 1b 3b 85 d8 bd
>1a60 1c 3b 85 d9 ac 67 3b c0
>1a68 00 f0 0d e6 d8 a5 d8 c9
>1a70 00 d0 02 e6 d9 88 d0 f3
>1a78 a5 d8 85 da a5 d9 e9 04
>1a80 85 db a9 7d a0 00 91 d8
>1a88 a9 58 91 da ea ea ea ea
>1a90 ea ea ea ea ea ea ea 60
>1a98 a2 00 ac 68 3b e8 e8 88
>1aa0 d0 fb bd 1b 3b 85 d8 bd
>1aa8 1c 3b 85 d9 ac 67 3b c0
>1ab0 00 f0 0d e6 d8 a5 d8 c9
>1ab8 00 d0 02 e6 d9 88 d0 f3
>1ac0 a0 00 b1 d8 60 ea ea ea
>1ac8 ea ea ea ea ea ea a2 00
>1ad0 ac 50 3b e8 e8 88 d0 fb
>1ad8 bd 1b 3b 85 d8 bd 1c 3b
>1ae0 85 d9 ac 4f 3b c0 00 f0
>1ae8 0d e6 d8 a5 d8 c9 00 d0
>1af0 02 e6 d9 88 d0 f3 a0 00
>1af8 b1 d8 c9 7d d0 03 20 3e
>1b00 2a 4c ae 15 ad 11 ff 2a
>1b08 2a 2a 2a b0 0b ad 11 ff
>1b10 09 10 8d 11 ff ea ea 60
>1b18 ad 11 ff 29 07 8d 11 ff
>1b20 ea ea 60 ea ea ea ad 6b
>1b28 3b c9 01 f0 03 ea ea 60
>1b30 ad 68 3b cd 6a 3b d0 03
>1b38 4c 69 1b ac 6a 3b a2 00
>1b40 e8 e8 88 d0 fb bd 1b 3b
>1b48 85 d8 bd 1c 3b 85 d9 ac
>1b50 69 3b c0 00 f0 0d e6 d8
>1b58 a5 d8 c9 00 d0 02 e6 d9
>1b60 88 d0 f3 a0 00 a9 20 91
>1b68 d8 ce 6a 3b ad 6a 3b c9
>1b70 01 d0 08 a9 00 8d 6b 3b
>1b78 ea ea 60 a2 00 ac 6a 3b
>1b80 e8 e8 88 d0 fb bd 1b 3b
>1b88 85 d8 bd 1c 3b 85 d9 ac
>1b90 69 3b c0 00 f0 0d e6 d8
>1b98 a5 d8 c9 00 d0 02 e6 d9
>1ba0 88 d0 f3 a0 00 b1 d8 c9
>1ba8 20 f0 03 4c e7 1b ac 6a
>1bb0 3b a2 00 e8 e8 88 d0 fb
>1bb8 bd 1b 3b 85 d8 bd 1c 3b
>1bc0 85 d9 ac 69 3b c0 00 f0
>1bc8 0d e6 d8 a5 d8 c9 00 d0
>1bd0 02 e6 d9 88 d0 f3 a0 00
>1bd8 a9 7e 91 d8 a5 d9 e9 04
>1be0 85 d9 a9 7f 91 d8 60 a2
>1be8 47 8e 11 ff ea ea ea c9
>1bf0 78 d0 09 a0 00 a9 79 91
>1bf8 d8 4c d9 1d c9 79 d0 09
>1c00 a0 00 a9 7a 91 d8 4c d9
>1c08 1d c9 7a d0 09 a9 7b a0
>1c10 00 91 d8 4c d9 1d c9 7b
>1c18 d0 09 a9 20 a0 00 91 d8
>1c20 4c d9 1d c9 7f f0 03 4c
>1c28 44 24 ce 00 3b ad 00 3b
>1c30 c9 00 d0 03 4c 3a 1e ac
>1c38 04 3b a2 00 e8 e8 88 d0
>1c40 fb bd 1b 3b 85 d8 bd 1c
>1c48 3b 85 d9 ac 03 3b c0 00
>1c50 f0 0d e6 d8 a5 d8 c9 00
>1c58 d0 02 e6 d9 88 d0 f3 a0
>1c60 00 a9 20 91 d8 a2 00 bd
>1c68 05 3b 9d 51 3b e8 e0 16
>1c70 d0 f5 a2 00 bd 51 3b 9d
>1c78 03 3b e8 e0 16 d0 f5 a9
>1c80 00 8d 6b 3b a2 fe ac 00
>1c88 3b e8 e8 88 d0 fb bd 03
>1c90 3b 8d 4f 3b bd 04 3b 8d
>1c98 50 3b 4c 06 1e ad 01 3b
>1ca0 c9 00 f0 0b ee 4f 3b ad
>1ca8 4f 3b c9 28 d0 0d 60 ce
>1cb0 4f 3b ad 4f 3b c9 ff d0
```

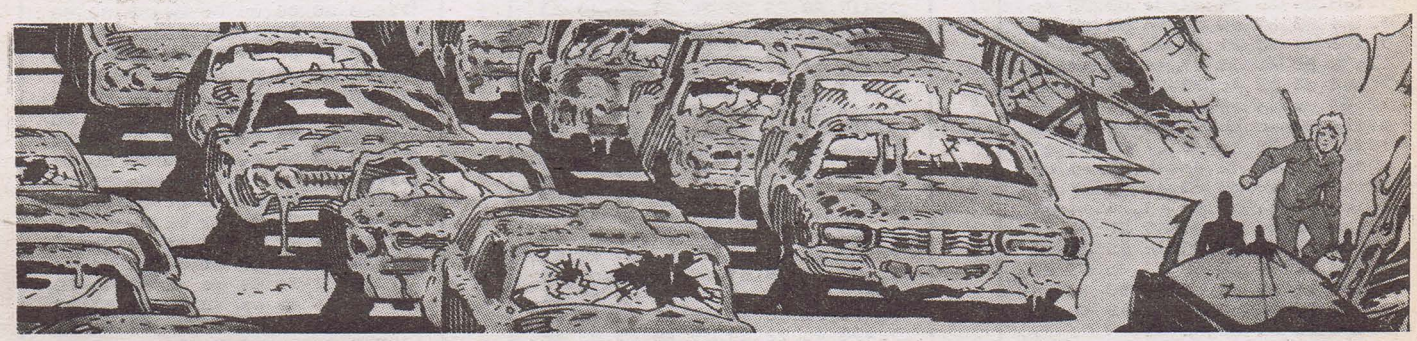
```
>1cb8 02 60 ea a2 00 ac 50 3b
>1cc0 e8 e8 88 d0 fb bd 1b 3b
>1cc8 85 d8 bd 1c 3b 85 d9 ac
>1cd0 4f 3b e6 d8 c9 00 d0 02
>1cd8 e6 d9 88 d0 f5 a9 78 a0
>1ce0 00 91 d8 a5 d9 e9 04 85
>1ce8 d9 a9 05 ea ea 91 d8 ea
>1cf0 ad 0b 0c c9 39 f0 04 ee
>1cf8 0b 0c 60 a9 30 8d 0b 0c
>1d00 ea ad 0a 0c c9 39 f0 06
>1d08 ee 0a 0c ea ea 60 a9 30
>1d10 8d 0a 0c ad 09 0c c9 39
>1d18 f0 06 ee 09 0c ea ea 60
>1d20 a9 30 8d 09 0c ad 08 0c
>1d28 f0 06 ee 08 0c ea ea 60
>1d30 a9 30 8d 08 0c 60 ce 68
>1d38 3b ad 68 3b c9 14 d0 05
>1d40 a9 15 8d 68 3b 20 98 1a
>1d48 c9 7f d0 03 20 3e 2a c9
>1d50 20 d0 03 4c 50 1a ee 68
>1d58 3b ea ea ea ea ea 4c 50
>1d60 1a ee 68 3b ad 68 3b c9
>1d68 19 d0 05 a9 18 8d 68 3b
>1d70 20 98 1a c9 7f d0 03 20
>1d78 3e 2a c9 20 d0 03 4c 50
>1d80 1a ce 68 3b 4c 50 1a ce
>1d88 67 3b ad 67 3b c9 ff d0
>1d90 05 a9 00 8d 67 3b 20 98
>1d98 1a c9 7f d0 03 20 3e 2a
>1da0 c9 20 d0 03 4c 50 1a ee
>1da8 67 3b 4c 50 1a ea ea ea
>1db0 ee 67 3b ad 67 3b c9 28
>1db8 d0 05 a9 27 8d 67 3b 20
>1dc0 98 1a c9 7f d0 03 20 3e
>1dc8 2a c9 20 d0 03 4c 50 1a
>1dd0 ce 67 3b 4c 50 1a ea ea
>1dd8 ea a9 00 8d 6b 3b ad 0d
>1de0 0c c9 39 f0 04 ee 0d 0c
>1de8 60 a9 30 8d 0d 0c ad 0c
>1df0 0c c9 39 f0 04 ee 0c 0c
>1df8 60 a9 30 8d 0c 0c 4c f0
>1e00 1c ea ea ea ea ea ae 50
>1e08 3b e0 18 d0 03 4c f0 1c
>1e10 ad 01 3b c9 00 f0 10 ee
>1e18 4f 3b ad 4f 3b c9 28 d0
>1e20 03 4c f0 1c 4c ac 1f ce
>1e28 4f 3b ad 4f 3b c9 ff d0
>1e30 03 4c f0 1c 4c ac 1f ea
>1e38 ea ea a2 00 ac 04 3b e8
>1e40 e8 88 d0 fb bd 1b 3b 85
>1e48 d8 bd 1c 3b 85 d9 ac 03
>1e50 3b c0 00 f0 0d e6 d8 a5
>1e58 d8 c9 00 d0 02 e6 d9 88
>1e60 d0 f3 a0 00 a9 20 91 d8
>1e68 a9 0c 8d 00 3b a9 00 aa
>1e70 9d 03 3b 69 01 e8 e8 e0
>1e78 18 d0 f5 a9 02 a2 01 9d
>1e80 03 3b e8 e8 e0 19 d0 f7
>1e88 ee 70 3b 4c f0 1c ad 12
>1e90 0c c9 20 f0 08 a9 20 8d
>1e98 12 0c 4c bb 1e ad 11 0c
>1ea0 c9 20 f0 08 a9 20 8d 11
>1ea8 0c 4c bb 1e ad 10 0c c9
>1eb0 20 d0 03 4c 8f 25 a9 20
>1eb8 8d 10 0c a2 00 ac 68 3b
>1ec0 e8 e8 88 d0 fb bd 1b 3b
>1ec8 85 d8 85 da bd 1c 3b 85
>1ed0 d9 85 db ac 67 3b c0 00
>1ed8 f0 11 e6 d8 e6 da a5 d8
>1ee0 c9 00 d0 04 e6 d9 e6 db
>1ee8 88 d0 ef a5 db e9 04 85
>1ef0 db a9 00 91 da 69 01 c9
>1ef8 ff d0 f8 a9 47 8d 11 ff
>1f00 a9 56 a0 00 91 d8 20 9b
>1f08 1f a9 57 a0 00 91 d8 20
>1f10 9b 1f a9 58 a0 00 91 d8
>1f18 20 9b 1f a9 59 a0 00 91
>1f20 d8 20 9b 1f a9 20 a0 00
>1f28 91 d8 a9 14 8d 67 3b a9
>1f30 18 8d 68 3b a9 07 8d 11
>1f38 ff a9 50 85 d8 a9 0c 85
>1f40 d9 a0 00 b1 d8 c9 20 f0
>1f48 24 c9 79 d0 03 4c 69 1f
>1f50 c9 7a d0 03 4c 69 1f c9
>1f58 7b d0 03 4c 69 1f c9 7f
>1f60 d0 04 a9 20 91 d8 4c 6d
```

```
>1f68 1f a9 78 91 d8 c8 d0 d3
>1f70 e6 d9 a5 d9 c9 10 d0 cb
>1f78 a9 0c 8d 00 3b a9 00 aa
>1f80 9d 03 3b 69 01 e8 e8 e0
>1f88 18 d0 f5 a9 02 a2 01 9d
>1f90 03 3b e8 e8 e0 19 d0 f7
>1f98 4c 54 2a a2 00 a0 00 c8
>1fa0 d0 fd e8 e0 e0 d0 f8 60
>1fa8 ea ea ea ea a2 00 ac 50
>1fb0 3b e8 e8 88 d0 fb bd 1b
>1fb8 3b 85 d8 bd 1c 3b 85 d9
>1fc0 ac 4f 3b c0 00 f0 0d e6
>1fc8 d8 a5 d8 c9 00 d0 02 e6
>1fd0 d9 88 d0 f3 a0 00 a9 78
>1fd8 91 d8 a5 d9 e9 04 85 d9
>1fe0 a9 02 91 d8 4c f0 1c ea
>1fe8 ea ea ea ea ea ea ea ea
>1ff0 a0 00 c8 d0 fd ee 6c 3b
>1ff8 ee 6d 3b ee 6e 3b ad 6c
>2000 3b c9 10 d0 08 20 6a 19
>2008 a9 00 8d 6c 3b ad 6d 3b
>2010 c9 10 d0 08 20 26 1b a9
>2018 00 8d 6d 3b ad 6e 3b cd
>2020 8b 3b ea ea d0 08 20 95
>2028 13 a9 00 8d 6e 3b ad 70
>2030 3b c9 05 d0 14 a9 00 8d
>2038 70 3b ee 6f 3b ad 6f 3b
>2040 c9 05 d0 05 a9 04 8d 6f
>2048 3b 4c 44 2a ea ea ad 71
>2050 3b ea c9 10 f0 03 4c a9
>2058 20 a9 00 8d 71 3b ad 6f
>2060 3b c9 00 d0 03 4c a9 20
>2068 c9 01 d0 03 20 af 20 c9
>2070 02 d0 06 20 af 20 20 ce
>2078 20 c9 03 d0 09 20 af 20
>2080 20 ce 20 20 ed 20 c9 04
>2088 d0 0c 20 af 20 20 ce 20
>2090 20 ed 20 20 0c 21 c9 05
>2098 d0 0f 20 af 20 20 ce 20
>20a0 20 ed 20 20 0c 21 20 44
>20a8 24 4c f0 1f ea ea ea ad
>20b0 72 3b ae 73 3b 8d 7a 3b
>20b8 8e 7b 3b 20 9c 23 ad 7a
>20c0 3b ae 7b 3b 8d 72 3b 8e
>20c8 73 3b 4c 27 21 60 ad 74
>20d0 3b ae 75 3b 8d 7a 3b 8e
>20d8 7b 3b 20 b8 23 ad 7a 3b
>20e0 ae 7b 3b 8d 74 3b 8e 75
>20e8 3b 4c 27 21 60 ad 76 3b
>20f0 ae 77 3b 8d 7a 3b 8e 7b
>20f8 3b 20 d4 23 ad 7a 3b ae
>2100 7b 3b 8d 76 3b 8e 77 3b
>2108 4c 27 21 60 ad 78 3b ae
>2110 79 3b 8d 7a 3b 8e 7b 3b
>2118 20 f0 23 ad 7a 3b ae 7b
>2120 3b 8d 78 3b 8e 79 3b a9
>2128 00 60 ea ea ea a2 00 ac
>2130 7b 3b e8 e8 88 d0 fb bd
>2138 1b 3b 85 d8 bd 1c 3b 85
>2140 d9 ac 7a 3b c0 00 f0 0d
>2148 e6 d8 a5 d8 c9 00 d0 02
>2150 e6 d9 88 d0 f3 a0 00 a9
>2158 20 91 d8 ea ea ea ad 7c
>2160 3b c9 01 f0 03 4c 63 22
>2168 ee 7a 3b ad 7a 3b c9 28
>2170 d0 5d a9 27 8d 7a 3b ad
>2178 7d 3b c9 01 d0 2a ee 7b
>2180 3b ad 7b 3b c9 19 d0 18
>2188 a9 18 8d 7b 3b a9 00 8d
>2190 7c 3b 8d 7d 3b ce 7b 3b
>2198 ea ea ea ea ea 4c 60 23
>21a0 a9 00 8d 7c 3b 4c 0c 24
>21a8 ce 7b 3b ad 7b 3b c9 01
>21b0 d0 15 a9 02 8d 7b 3b a9
>21b8 00 8d 7c 3b a9 01 8d 7d
>21c0 3b ee 7b 3b 4c 60 23 a9
>21c8 00 8d 7c 3b 4c 60 23 ad
>21d0 7b 3b 2a aa bd 1b 3b 85
>21d8 d8 e8 bd 1b 3b 85 d9 ad
>21e0 7a 3b c9 00 f0 12 e9 01
>21e8 8d 4e 3b a5 d8 6d 4e 3b
>21f0 c5 d8 b0 02 e6 d9 85 d8
>21f8 a0 00 b1 d8 c9 7d d0 03
>2200 20 3e 2a c9 20 d0 03 4c
>2208 60 23 ad 7d 3b c9 01 d0
>2210 28 ce 7a 3b ee 7b 3b ad
>2218 7b 3b c9 19 d0 13 a9 18
>2220 8d 7b 3b a9 00 8d 7c 3b
>2228 8d 7d 3b ce 7b 3b 4c 60
>2230 23 a9 00 8d 7c 3b 4c 60
>2238 23 ce 7a 3b ce 7b 3b ad
>2240 7b 3b c9 01 d0 15 a9 02
>2248 8d 7b 3b a9 00 8d 7c 3b
>2250 a9 01 8d 7d 3b ee 7b 3b
>2258 4c 60 23 a9 00 8d 7c 3b
>2260 4c 60 23 ce 7a 3b ad 7a
>2268 3b c9 ff d0 5d a9 00 8d
>2270 7a 3b ad 7d 3b c9 01 d0
>2278 2a ee 7b 3b ad 7b 3b c9
>2280 19 d0 18 a9 18 8d 7b 3b
>2288 a9 01 8d 7c 3b a9 00 8d
>2290 7d 3b ce 7b 3b ea ea ea
>2298 4c 60 23 a9 01 8d 7c 3b
>22a0 4c 0c 24 ce 7b 3b ad 7b
>22a8 3b c9 01 d0 15 a9 02 8d
>22b0 7b 3b a9 01 8d 7c 3b a9
>22b8 01 8d 7d 3b ee 7b 3b 4c
>22c0 60 23 a9 01 8d 7c 3b 4c
>22c8 60 23 ad 7b 3b 2a aa bd
>22d0 1b 3b 85 d8 e8 bd 1b 3b
>22d8 85 d9 ad 7a 3b c9 00 f0
>22e0 78 e9 01 8d 4e 3b a5 d8
>22e8 6d 4e 3b c5 d8 b0 02 e6
>22f0 d9 85 d8 a0 00 b1 d8 c9
>22f8 7d d0 03 20 3e 2a c9 20
>2300 d0 03 4c 60 23 ad 7d 3b
>2308 c9 01 d0 2a ee 7a 3b ee
>2310 7b 3b ad 7b 3b c9 19 d0
>2318 15 a9 18 8d 7b 3b a9 01
>2320 8d 7c 3b a9 00 8d 7d 3b
>2328 ce 7b 3b 4c 60 23 a9 01
>2330 8d 7c 3b 4c 60 23 ee 7a
>2338 3b ce 7b 3b ad 7b 3b c9
>2340 01 d0 15 a9 02 8d 7b 3b
>2348 a9 01 8d 7c 3b a9 01 8d
>2350 7d 3b ee 7b 3b 4c 60 23
>2358 a9 01 8d 7c 3b ea ea ea
>2360 a2 00 ac 7b 3b e8 e8 88
>2368 d0 fb bd 1b 3b 85 d8 bd
>2370 1c 3b 85 d9 ac 7a 3b c0
>2378 00 f0 0d e6 d8 a5 d8 c9
>2380 00 d0 02 e6 d9 88 d0 f3
>2388 a0 00 a9 5b 91 d8 a5 d9
>2390 e9 04 85 d9 a9 7f 91 d8
>2398 ea ea ea 60 ad 7e 3b ae
>23a0 7f 3b 8d 7c 3b 8e 7d 3b
>23a8 20 2d 21 ad 7c 3b ae 7d
>23b0 3b 8d 7e 3b 8e 7f 3b 60
>23b8 ad 80 3b ae 81 3b 8d 7c
>23c0 3b 8e 7d 3b 20 2d 21 ad
>23c8 7c 3b ae 7d 3b 8d 80 3b
>23d0 8e 81 3b 60 ad 82 3b ae
>23d8 83 3b 8d 7c 3b 8e 7d 3b
>23e0 20 2d 21 ad 7c 3b ae 7d
>23e8 3b 8d 82 3b 8e 83 3b 60
>23f0 ad 84 3b ae 85 3b 8d 7c
>23f8 3b 8e 7d 3b 20 2d 21 ad
>2400 7c 3b ae 7d 3b 8d 84 3b
>2408 8e 85 3b 60 a2 00 ac 50
>2410 3b e8 e8 88 d0 fb bd 1b
>2418 3b 85 d8 bd 1c 3b 85 d9
>2420 ac 4f 3b c0 00 f0 0d e6
>2428 d8 a5 d8 c9 00 d0 02 e6
>2430 d9 88 d0 f3 a0 00 b1 d8
>2438 c9 7d d0 03 20 3e 2a 4c
>2440 60 23 ea ea c9 5b f0 05
>2448 ea ea ea ea 60 a9 00 8d
>2450 6b 3b ad 69 3b cd 72 3b
>2458 d0 0b ad 6a 3b cd 73 3b
>2460 d0 03 4c a1 24 ad 69 3b
>2468 cd 74 3b d0 0b ad 6a 3b
>2470 cd 75 3b d0 03 4c ba 24
>2478 ad 69 3b cd 76 3b d0 0b
>2480 ad 6a 3b cd 77 3b d0 03
>2488 4c d3 24 ad 69 3b cd 78
>2490 3b d0 0d ad 6a 3b cd 79
>2498 3b d0 05 4c ec 24 ea ea
>24a0 60 ad 72 3b ae 73 3b 8d
>24a8 7a 3b 8e 7b 3b a9 00 a0
>24b0 10 8d 72 3b 8c 73 3b 4c
>24b8 07 25 ad 74 3b ac 75 3b
>24c0 8d 7a 3b 8c 7b 3b a9 00
>24c8 a0 10 8d 74 3b 8c 75 3b
>24d0 4c 07 25 ad 76 3b ac 77
>24d8 3b 8d 7a 3b 8c 7b 3b a9
>24e0 00 a0 10 8d 76 3b 8c 77
>24e8 3b 4c 07 25 ad 78 3b ac
>24f0 79 3b 8d 7a 3b 8c 7b 3b
>24f8 a9 00 a0 10 8d 78 3b 8c
>2500 79 3b ea ea ea ea ea a2
>2508 00 ac 7b 3b e8 e8 88 d0
>2510 fb bd 1b 3b 85 d8 bd 1c
>2518 3b 85 d9 ac 7a 3b c0 00
>2520 f0 0d e6 d8 a5 d8 c9 00
>2528 d0 02 e6 d9 88 d0 f3 a0
>2530 00 a9 20 91 d8 ea ea ea
>2538 4c f0 1c ea ea ea ea ea
>2540 30 30 34 30 37 35 00 00
>2548 ad 40 25 cd 08 0c 90 3f
>2550 d0 3a ad 41 25 cd 09 0c
>2558 90 35 d0 30 ad 42 25 cd
>2560 0a 0c 90 2b d0 26 ad 43
>2568 25 cd 0a 0c 90 21 d0 1c
>2570 ad 44 25 cd 0b 0c 90 17
>2578 d0 12 ad 45 25 cd 0c 0c
>2580 90 0d d0 08 ad 46 25 cd
>2588 0d 0c 90 03 4c 9c 25 a2
>2590 00 bd 08 0c 9d 40 25 e8
>2598 e0 06 d0 f5 20 00 26 a2
>25a0 00 bd d5 25 20 d2 ff e8
>25a8 e0 25 d0 f5 ea ea a9 00
>25b0 aa a8 8d 4d 3b ea ea ea
>25b8 ea ea ea ea e8 d0 f6 c8
>25c0 d0 f3 ee 4d 3b ad 4d 3b
>25c8 c9 02 d0 e9 ea ea ea ea
>25d0 ea ea 4c 22 28 02 05 11
>25d8 11 11 11 11 11 11 11 11
>25e0 11 1d 1d 1d 1d 1d 1d 1d
>25e8 1d 1d 1d 1d 1d 1d 1d 47
>25f0 41 4d 45 20 4f 56 45 52
>25f8 1d 1d 1d 1d 1d 00 ea ea
>2600 a9 07 8d 11 ff a2 00 bd
>2608 40 25 9d 1e 0c e8 e0 06
>2610 d0 f5 60 ea ea ea 54 00
>2618 86 4c 13 05 0d 1d 1d 1d
>2620 1d 1d 1d 1d 1d 1d cf b7
>2628 b7 b7 b7 b7 b7 b7 b7 b7
>2630 b7 b7 b7 b7 b7 b7 b7 b7
>2638 b7 b7 b7 d0 0d 1d 1d 1d
>2640 1d 1d 1d 1d 1d 1d b4 20
>2648 20 20 20 43 52 41 5a 59
>2650 20 20 57 4f 52 4d 53 20
>2658 20 20 20 aa 0d 1d 1d 1d
>2660 1d 1d 1d 1d 1d 1d cc af
>2668 af af af af af af af af
>2670 af af af af af af af af
>2678 af af af da 0d 0d 0d 0d
>2680 1d 1d 1d 1d 1d 1d 1d 1d
>2688 1d 50 52 4f 47 52 41 4d
>2690 4d 45 44 20 42 59 20 46
>2698 52 41 4e 4b 20 4c 45 41
>26a0 0d 0d 1d 1d 1d 1d 43 4f
>26a8 50 59 52 49 47 48 54 20
>26b0 31 39 38 36 20 42 59 20
>26b8 54 52 4f 4e 49 43 20 56
>26c0 45 52 4c 41 47 0d af af
>26c8 af af af af af af af af
>26d0 af af af af af af af af
>26d8 af af af af af af af af
>26e0 af af af af af af af af
>26e8 af af af af af af 0d 0d
>26f0 20 20 20 20 20 20 20 46
>26f8 31 3d 53 50 45 45 44 20
>2700 20 20 20 20 20 20 20 20
>2708 20 46 32 3d 53 45 45 20
>2710 4b 45 59 53 20 20 20 0d
>2718 0d 20 20 20 50 52 45 53
>2720 53 20 4a 4f 59 23 31 20
>2728 46 49 52 45 20 4f 52 20
>2730 53 50 41 43 45 20 54 4f
>2738 20 50 4c 41 59 0d 0d 1d
>2740 1d 1d 1d 1d 1d 1d 1d 1d
>2748 1d 1d 53 50 45 45 44 0d
>2750 c3 c3 c3 c3 c3 c3 c3 c3
>2758 c3 c3 c3 c3 c3 c3 c3 c3
>2760 c3 c3 c3 c3 c3 c3 c3 c3
>2768 c3 c3 c3 c3 c3 c3 c3 c3
>2770 c3 c3 c3 c3 c3 c3 c3 c3
```

```
>2778 20 cf b7 b7 b7 b7 b7 b7
>2780 b7 b7 b7 b7 b7 b7 b7 b7
>2788 b7 b7 b7 b7 b7 b7 b7 b7
>2790 b7 b7 b7 b7 b7 b7 b7 b7
>2798 b7 b7 b7 b7 b7 b7 d0 0d
>27a0 20 b4 11 9d b4 11 9d b4
>27a8 11 9d b4 11 9d cc af af
>27b0 af af af af af af af af
>27b8 af af af af af af af af
>27c0 af af af af af af af af
>27c8 af af af af af af af af
>27d0 af af da 91 9d aa 91 9d
>27d8 aa 91 9d aa 91 9d aa 11
>27e0 9d 9d 9d 9d 9d 9d 9d 9d
>27e8 9d 9d 9d 9d 9d 9d 9d 9d
>27f0 9d 9d 9d 9d 9d 9d 9d 9d
>27f8 9d 20 20 20 20 ff ff 59
>2800 53 20 20 20 20 20 9d 9d
>2808 9d 9d 9d 9d 9d 9d 9d 9d
>2810 9d 9d 20 20 20 20 4b 45
>2818 59 53 20 20 20 20 9d 9d
>2820 9d 9d a9 93 20 d2 ff a9
>2828 00 8d 15 ff 8d 19 ff a9
>2830 1a a2 26 85 d8 86 d9 a0
>2838 00 b1 d8 20 d2 ff c8 c0
>2840 03 f0 09 c0 00 d0 f2 e6
>2848 d9 4c 39 28 a6 d9 e0 28
>2850 d0 e7 ad 8a 3b c9 00 f0
>2858 0b c9 01 f0 1c c9 02 f0
>2860 2d 4c a3 28 a0 00 b9 f7
>2868 28 20 d2 ff c8 c0 28 d0
>2870 f5 a9 30 8d 8b 3b 4c a3
>2878 28 a0 00 b9 1f 29 20 d2
>2880 ff c8 c0 28 d0 f5 a9 15
>2888 8d 8b 3b 4c a3 28 a0 00
>2890 b9 47 29 20 d2 ff c8 c0
>2898 28 d0 f5 a9 10 8d 8b 3b
>28a0 ea ea ea a5 c6 c9 40 f0
>28a8 fa c9 04 d0 06 20 da 28
>28b0 4c 5c 2a c9 05 d0 06 20
>28b8 6f 29 4c 52 28 c9 06 d0
>28c0 06 4c 52 28 4c 52 28 c9
>28c8 3c d0 03 4c 2d 2a c9 16
>28d0 f0 f9 4c 52 28 ea ea ea
>28d8 ea ea ee 8a 3b ad 8a 3b
>28e0 c9 03 d0 05 a9 00 8d 8a
```

```
>28e8 3b a2 00 a0 00 c8 d0 fd
>28f0 e8 e0 e0 d0 f8 60 ea 13
>28f8 11 11 11 11 11 11 11 11
>2900 11 11 11 11 11 11 11 11
>2908 1d 1d 1d 1d 1d 1d 1d 1d
>2910 1d 1d 1d 1d 1d 1d 1d 1d
>2918 1d 53 4c 4f 57 20 20 13
>2920 11 11 11 11 11 11 11 11
>2928 11 11 11 11 11 11 11 11
>2930 1d 1d 1d 1d 1d 1d 1d 1d
>2938 1d 1d 1d 1d 1d 1d 1d 1d
>2940 1d 46 41 53 54 20 20 13
>2948 11 11 11 11 11 11 11 11
>2950 11 11 11 11 11 11 11 11
>2958 1d 1d 1d 1d 1d 1d 1d 1d
>2960 1d 1d 1d 1d 1d 1d 1d 1d
>2968 1d 43 52 41 5a 59 20 a9
>2970 16 8d e5 07 a9 13 8d e6
>2978 07 a9 0d 8d e7 07 a9 1b
>2980 8d e8 07 a9 11 a0 00 a2
>2988 00 20 d2 ff c8 d0 fd e8
>2990 e4 04 d0 f5 a9 13 20 d2
>2998 ff a9 11 20 d2 ff 20 d2
>29a0 ff a0 00 a9 1d 20 d2 ff
>29a8 c8 c0 04 d0 f8 a9 4b 20
>29b0 d2 ff a9 45 20 d2 ff a9
>29b8 59 20 d2 ff a9 53 20 d2
>29c0 ff a2 00 bd ea 29 20 d2
>29c8 ff c9 0d d0 10 8a 48 a2
>29d0 00 a0 00 e8 d0 fd c8 c0
>29d8 e0 d0 f8 68 aa e8 e0 2b
>29e0 d0 e1 4c 1a 2a ea ea ea
>29e8 ea ea 11 11 11 0d 1d 1d
>29f0 1d 1d 1d 1d 53 20 0d 1d
>29f8 1d 1d 1d 1d 3a 20 3b 0d
>2a00 1d 1d 1d 1d 1d 1d 58 20
>2a08 0d 1d 1d 46 49 52 45 3d
>2a10 53 50 41 43 45 0d 0d 0d
>2a18 0d 0d a9 18 8d e5 07 a9
>2a20 00 8d e6 07 8d e7 07 a9
>2a28 27 8d e8 07 60 ea ea ea
>2a30 a9 93 20 d2 ff 20 e0 2a
>2a38 4c 0c 13 ea ea ea a9 ff
>2a40 8d 8c 3b 60 ad 8c 3b c9
>2a48 00 d0 06 ee 71 3b 4c 4e
>2a50 20 4c 8e 1e a9 00 8d 8c
```

```
>2a58 3b 4c 4e 20 ad 8a 3b c9
>2a60 00 d0 08 a9 30 8d 8b 3b
>2a68 4c 52 28 c9 01 d0 08 a9
>2a70 15 8d 8b 3b 4c 52 28 c9
>2a78 02 d0 05 a9 10 8d 8b 3b
>2a80 4c 52 28 ea ea ea a9 30
>2a88 8d 8b 3b a9 00 8d 8a 3b
>2a90 ea ea a2 00 bd a5 2a 9d
>2a98 b0 3e e8 e0 30 d0 f5 4c
>2aa0 22 28 ea ea ea 00 00 20
>2aa8 18 28 00 00 00 00 44 28
>2ab0 1c 38 44 00 00 62 66 3e
>2ab8 1c 7e 7e 56 c2 62 ee 7f
>2ac0 7e ff 7e 56 c2 03 03 03
>2ac8 03 03 03 ff ff 42 3c 7e
>2ad0 7e 7e 7e 3c 42 ea ea ea
>2ad8 ea ea ea ea ea ea ea ea
>2ae0 a9 0c 8d 00 3b a9 01 8d
>2ae8 01 3b 8d 02 3b a2 00 bd
>2af0 45 13 9d 03 3b e8 e0 4a
>2af8 d0 f5 a9 00 8d 52 3b a9
>2b00 14 a2 18 8d 67 3b 8e 68
>2b08 3b a9 00 8d 6b 3b a9 07
>2b10 8d 11 ff a9 00 8d 0e ff
>2b18 a9 ff 8d 0f ff a9 7e 8d
>2b20 10 ff a9 7d 8d 10 0c 8d
>2b28 11 0c 8d 12 0c a9 00 8d
>2b30 6c 3b 8d 6d 3b 8d 6e 3b
>2b38 8d 6f 3b 8d 70 3b 8d 71
>2b40 3b a9 00 a2 10 8d 72 3b
>2b48 8e 73 3b 8d 74 3b 8e 75
>2b50 3b 8d 76 3b 8e 77 3b 8d
>2b58 78 3b 8e 79 3b a9 01 8d
>2b60 7c 3b 8d 7d 3b a9 00 8d
>2b68 86 3b a9 20 8d 89 3b a9
>2b70 01 8d 7e 3b 8d 7f 3b 8d
>2b78 80 3b 8d 81 3b 8d 82 3b
>2b80 8d 83 3b 8d 84 3b 8d 85
>2b88 3b a2 00 bd 40 25 9d 1e
>2b90 0c e8 e0 06 d0 f5 ea ea
>2b98 ea ea ea ea ea ea ea ea
>2ba0 a9 00 8d 8c 3b ea ea ea
>2ba8 a2 00 a9 20 9d 5f 05 e8
>2bb0 e0 a0 d0 f8 ea ea ea 60
```



Convoy

Ein starkes Spiel

100% Maschinensprache garantieren flotte Action auf hoher See. Ihre Aufgabe ist es, feindliche Schiffe und U-Boote mit Ihren Torpedos zu versenken.

Eingegeben wird dieses Programm über den eingebauten Maschinensprachemonitor (M, RETURN).

Das Programm belegt den Speicherplatz von **$2000 - $3FFF.**

gestartet wird es mit **SYS 13056** oder aus dem Monitor heraus mit **G 3300.**

Die Spielgeschwindigkeit läßt sich übrigens verringern oder erhöhen, indem vor dem Programmstart in die Speicherstelle 13197 ein anderer Wert gepoket wird (Die Werte können Sie leicht durch ausprobieren herausbekommen).

Hier geht's los

```
>2000 a9 fd 8d 08 ff ad 08 ff
>2008 c9 7b d0 03 4c 85 20 c9
>2010 f7 d0 03 20 75 20 c9 77
>2018 d0 03 4c 85 20 c9 fb d0
>2020 03 20 65 20 c9 7f d0 03
>2028 4c 85 20 60 ea 20 ea ea
>2030 ea 60 a9 58 85 d8 a9 0f
>2038 85 d9 60 a9 40 a2 00 a4
>2040 d8 81 d8 e6 d8 a9 41 81
>2048 d8 e6 d8 a9 42 81 d8 84
>2050 d8 60 a9 20 a4 d8 a2 00
>2058 81 d8 e6 d8 81 d8 e6 d8
>2060 81 d8 84 d8 60 a6 d8 e0
>2068 48 d0 01 60 20 52 20 c6
>2070 d8 20 3b 20 60 a6 d8 e0
>2078 6d d0 01 60 20 52 20 e6
>2080 d8 20 3b 20 60 ae 05 31
>2088 e0 01 d0 03 4c 2b 20 a2
>2090 01 8e 05 31 a6 d8 e8 86
>2098 e4 a5 e4 38 e9 28 85 e4
>20a0 a5 d9 e9 00 85 e5 4c 2b
>20a8 20 ae 05 31 e0 01 f0 01
>20b0 60 a9 20 a2 00 81 e4 a5
>20b8 e4 38 e9 28 85 e4 a5 e5
>20c0 e9 00 85 e5 a1 e4 c9 20
>20c8 d0 07 a9 47 a2 00 81 e4
>20d0 60 ea c9 51 d0 03 4c 00
>20d8 21 c9 52 d0 03 4c 00 21
>20e0 c9 53 d0 03 4c 00 21 c9
>20e8 54 d0 03 4c 00 21 4c 0c
>20f0 21 ae 05 31 e0 01 f0 b9
>20f8 60 00 00 00 00 00 00 00
>2100 a2 00 8e 05 31 a2 00 a2
>2108 00 81 e4 60 a4 da a6 e4
>2110 e4 da d0 03 4c 00 27 e6
>2118 da e4 da d0 03 4c 00 27
>2120 e6 da e4 da d0 03 4c 00
>2128 27 e6 da e4 da d0 03 4c
>2130 00 27 84 da ea a4 dc a6
>2138 e4 e4 dc d0 03 4c 19 27
>2140 e6 dc e4 dc d0 03 4c 19
>2148 27 e6 dc e4 dc d0 03 4c
>2150 19 27 e6 dc e4 dc d0 03
>2158 4c 19 27 84 dc ea ea a4
>2160 de a6 e4 e4 de d0 03 4c
>2168 33 27 e6 de e4 de d0 03
>2170 4c 33 27 e6 de e4 de d0
>2178 03 4c 33 27 e6 de e4 de
>2180 d0 03 4c 33 27 84 de ea
>2188 ea a4 e0 a6 e4 e4 e0 d0
>2190 03 4c 4b 27 e6 e0 e4 e0
>2198 d0 03 4c 4b 27 c6 e0 e4
>21a0 e0 d0 03 4c 4b 27 e6 e0
>21a8 e4 e0 d0 03 4c 4b 27 84
>21b0 e0 ea ea a4 e2 a6 e4 e4
>21b8 e2 d0 03 4c 65 27 e6 e2
>21c0 e4 e2 d0 03 4c 65 27 c6
>21c8 e2 e4 e2 d0 03 4c 65 27
>21d0 e6 e2 e4 e2 d0 03 4c 65
>21d8 27 84 e2 60 60 a9 00 8d
>21e0 0b 31 8d 0c 31 8d 0d 31
>21e8 8d 0e 31 8d 0f 31 8d 10
>21f0 31 8d 11 31 60 ea ea ea
>21f8 22 46 0a 99 22 20 53 ff
>2200 a9 20 a2 00 a4 da 81 da
>2208 e6 da 81 da e6 da 81 da
>2210 e6 da 81 da 84 da 60 a9
>2218 43 a2 00 a4 da 81 da a9
>2220 44 e6 da 81 da a9 45 e6
>2228 da 81 da a9 46 e6 da 81
>2230 da 84 da 60 a9 20 a2 00
>2238 a4 dc 81 dc e6 dc 81 dc
>2240 e6 dc 81 dc e6 dc 81 dc
>2248 84 dc 60 a9 48 a2 00 a4
>2250 dc 81 dc a9 49 e6 dc 81
>2258 dc a9 4a e6 dc 81 dc a9
>2260 4b e6 dc 81 dc 84 dc 60
>2268 00 a9 20 a2 00 a4 de 81
>2270 de e6 de 81 de e6 de 81
>2278 de e6 de 81 de 84 de 60
>2280 a9 48 a2 00 a4 de 81 de
>2288 a9 49 e6 de 81 de a9 4a
>2290 e6 de 81 de a9 4b e6 de
>2298 81 de 84 de 60 00 a9 20
>22a0 a2 00 a4 e0 81 e0 e6 e0
>22a8 81 e0 84 e0 60 a9 4c a2
>22b0 00 a4 e0 81 e0 a9 4d e6
>22b8 e0 81 e0 84 e0 60 a9 20
>22c0 a2 00 a4 e2 81 e2 e6 e2
>22c8 81 e2 84 e2 60 a9 4c a2
>22d0 00 a4 e2 81 e2 a9 4d e6
>22d8 e2 81 e2 84 e2 60 a9 00
>22e0 8d 11 ff 8d 0e ff 8d 0f
>22e8 ff 0d 12 ff 0d 10 ff 4c
>22f0 00 28 00 00 00 00 00 00
>22f8 00 00 00 00 00 00 00 00
>2300 ae 00 31 e0 01 f0 01 60
>2308 a4 da c0 90 f0 09 20 00
>2310 22 c6 da 20 17 22 60 a9
>2318 00 8d 00 31 20 00 22 60
>2320 ae 01 31 e0 01 f0 01 60
>2328 a4 dc c0 64 f0 09 20 34
>2330 22 e6 dc 20 4b 22 60 20
>2338 34 22 a9 00 8d 01 31 60
>2340 ae 02 31 e0 01 f0 01 60
>2348 a4 de c0 dc f0 09 20 69
>2350 22 e6 de 20 80 22 60 20
>2358 69 22 a9 00 8d 02 31 60
>2360 ae 03 31 e0 01 f0 01 60
>2368 a4 e0 c0 30 f0 09 20 9e
>2370 22 c6 e0 20 ad 22 60 20
>2378 9e 22 a9 00 8d 03 31 60
>2380 ae 04 31 e0 01 f0 01 60
>2388 a4 e2 c0 3e f0 09 20 be
>2390 22 e6 e2 20 cd 22 60 20
>2398 be 22 a9 00 8d 04 31 60
>23a0 ad 11 31 8d ad 0f ad 10
>23a8 31 8d ae 0f ad 0f 31 8d
>23b0 af 0f ad 0e 31 8d b0 0f
>23b8 ad 0d 31 8d b1 0f ad 0c
>23c0 31 8d b2 0f ad 0b 31 8d
>23c8 b3 0f 60 a2 00 a0 00 e8
>23d0 4c d3 23 ee 15 ff ee 19
>23d8 ff e0 f0 d0 f2 ee 15 ff
>23e0 ee 19 ff c8 c0 f0 d0 e7
>23e8 60 00 00 00 00 00 00 00
>23f0 00 00 00 00 00 00 00 00
>23f8 00 00 00 00 00 00 00 00
>2400 20 00 20 20 00 23 20 00
>2408 25 20 00 20 20 1f 25 20
>2410 20 23 20 3e 25 20 40 23
>2418 20 00 20 20 60 23 20 5c
>2420 25 20 7a 25 20 80 23 20
>2428 7e 27 20 de 22 20 a9 20
>2430 20 00 25 20 a0 23 20 2e
>2438 29 20 00 30 20 5e 29 20
>2440 5b 30 20 8e 29 20 60 26
>2448 20 80 33 20 b3 24 20 58
>2450 24 4c e3 24 4c 00 24 00
>2458 ae 1b 31 e0 01 d0 49 ae
>2460 1c 31 86 e6 ae 1d 31 86
>2468 e7 a2 00 a9 20 81 e6 a5
>2470 e6 18 69 28 85 e6 a5 e7
>2478 69 00 85 e7 a1 e6 c9 20
>2480 d0 11 a9 4b 81 e6 a5 e6
>2488 8d 1c 31 a5 e7 8d 1d 31
>2490 4c b2 24 c9 40 d0 03 4c
>2498 00 32 c9 41 d0 03 4c 00
>24a0 32 c9 42 d0 03 4c 00 32
>24a8 c9 0a f0 01 60 a2 00 8e
>24b0 1b 31 60 ad 03 31 c9 01
>24b8 d0 28 ae 1b 31 e0 01 f0
>24c0 21 ae 00 ff e0 01 f0 0b
>24c8 e0 05 f0 07 e0 aa d0 03
>24d0 4c e2 24 a2 01 8e 1b 31
>24d8 a6 e0 8e 1c 31 a6 e1 8e
>24e0 1d 31 60 20 ce 32 20 73
>24e8 32 4c 54 24 00 00 00 00
>24f0 00 00 00 00 00 00 00 00
>24f8 00 00 00 00 00 00 00 00
>2500 ae 00 31 e0 01 d0 01 60
>2508 ae 00 ff e0 04 f0 01 60
>2510 a2 01 8e 00 31 a2 b4 86
>2518 da a2 0d 86 db 60 ea ae
>2520 01 31 e0 01 d0 01 60 ae
>2528 00 ff e0 06 f0 01 60 a2
>2530 01 8e 01 31 a2 40 86 dc
>2538 a2 0d 86 dd 60 00 ae 02
>2540 31 e0 01 d0 01 60 ae 00
>2548 ff e0 08 f0 01 60 a2 01
>2550 8e 02 31 a2 b8 86 de a2
>2558 0d 86 df 60 ae 03 31 e0
>2560 01 d0 01 60 ae 00 ff e0
>2568 05 f0 01 60 a2 01 8e 03
>2570 31 a2 54 86 e0 a2 0e 86
>2578 e1 60 ae 04 31 e0 01 d0
>2580 01 60 ae 00 ff e0 01 f0
>2588 01 60 a2 01 8e 04 31 a2
>2590 18 86 e2 a2 0d 86 e3 60
>2598 20 32 20 20 dd 21 20 3b
>25a0 20 a2 00 8e 12 31 8e 15
>25a8 31 8e 18 31 8e 05 31 60
>25b0 20 98 25 20 00 25 20 00
>25b8 23 20 1f 25 20 20 23 20
>25c0 3e 25 20 40 23 20 5c 25
>25c8 20 60 23 20 7a 25 20 80
>25d0 23 20 de 22 20 80 33 a5
>25d8 c6 c9 3c d0 d6 4c 3b 33
>25e0 00 00 00 00 00 00 00 00
>25e8 00 00 00 00 00 00 00 00
>25f0 00 00 00 00 00 00 00 00
>25f8 00 00 00 00 00 00 00 00
>2600 a9 00 e0 00 f0 0a ca ee
>2608 0b 31 20 11 26 4c 02 26
>2610 60 ac 0b 31 c0 0a f0 01
>2618 60 8d 0b 31 ee 0c 31 ac
>2620 0c 31 c0 0a f0 01 60 8d
>2628 0c 31 ee 0d 31 ac 0d 31
>2630 c0 0a f0 01 60 8d 0d 31
>2638 ee 0e 31 ac 0e 31 c0 0a
>2640 f0 01 60 8d 0e 31 ee 0f
>2648 31 c0 0a f0 01 60 8d 0f
>2650 31 ee 10 31 c0 0a f0 01
>2658 60 8d 10 31 ee 11 31 60
>2660 ae 18 31 e0 01 d0 49 ae
>2668 19 31 86 e6 ae 1a 31 86
>2670 e7 a2 00 a9 20 81 e6 a5
>2678 e6 18 69 28 85 e6 a5 e7
>2680 69 00 85 e7 a1 e6 c9 20
>2688 d0 11 a9 4b 81 e6 a5 e6
>2690 8d 19 31 a5 e7 8d 1a 31
>2698 4c ba 26 c9 40 d0 03 4c
>26a0 00 32 c9 41 d0 03 4c 00
>26a8 32 c9 42 d0 03 4c 00 32
>26b0 c9 0a f0 01 60 a2 00 8e
>26b8 18 31 60 00 00 00 00 00
>26c0 00 00 00 00 00 00 00 00
>26c8 00 00 00 00 00 00 00 00
>26d0 00 00 00 00 00 00 00 00
>26d8 00 00 00 00 00 00 00 00
>26e0 00 00 00 00 00 00 00 00
>26e8 00 00 00 00 00 00 00 00
>26f0 00 00 00 00 00 00 00 00
>26f8 00 00 00 00 00 00 00 00
>2700 a2 00 8e 05 31 8e 00 31
>2708 a9 4f 81 da a2 01 8e 06
>2710 31 84 da a2 02 4c 00 26
>2718 ea a2 00 8e 05 31 8e 01
>2720 31 a9 4f 81 dc a2 01 8e
>2728 07 31 84 dc a2 03 4c 00
>2730 26 ea ea a2 00 8e 05 31
>2738 8e 02 31 a9 4f 81 de a2
>2740 01 8e 08 31 84 de a2 01
>2748 4c 00 26 ea a2 00 8e 05
>2750 31 8e 03 31 a9 4f 81 e0
>2758 a2 01 8e 09 31 84 e0 a2
>2760 05 4c 00 26 ea a2 00 8e
>2768 05 31 8e 04 31 a9 4f 81
>2770 e2 a2 01 8e 0a 31 84 e2
>2778 a2 08 4c 00 26 ea a0 00
>2780 ae 06 31 e0 02 f0 03 4c
>2788 90 27 8c 06 31 20 00 22
>2790 ae 07 31 e0 02 f0 03 4c
>2798 a0 27 8c 07 31 20 34 22
>27a0 ae 08 31 e0 02 f0 03 4c
>27a8 b0 27 8c 08 31 20 69 22
>27b0 ae 09 31 e0 02 f0 03 4c
>27b8 c0 27 8c 09 31 20 9e 22
>27c0 ae 0a 31 e0 02 f0 03 4c
>27c8 d0 27 8c 0a 31 20 be 22
>27d0 60 00 00 00 00 00 00 00
>27d8 00 00 00 00 00 00 00 00
```

```
>27e0 ff ff ff ff ff ff ff ff
>27e8 ff ff ff ff ff ff ff ff
>27f0 00 00 00 00 00 00 00 00
>27f8 00 00 00 00 00 00 00 00
>2800 ae 05 31 e0 01 d0 19 a9
>2808 44 8d 11 ff a9 ff 8d 0f
>2810 ff a9 02 8d 10 ff a9 02
>2818 8d 10 ff a5 e4 8d 0e ff
>2820 ae 00 31 e0 01 d0 10 ad
>2828 11 ff 09 13 8d 11 ff ad
>2830 12 ff 09 00 8d 12 ff ae
>2838 01 31 e0 01 d0 18 ad 11
>2840 ff 09 13 8d 11 ff ad 12
>2848 ff 09 01 8d 12 ff ad 10
>2850 ff 09 00 8d 10 ff ad 02
>2858 31 c9 01 d0 10 ad 11 ff
>2860 09 14 8d 11 ff ad 12 ff
>2868 09 00 8d 12 ff ad 03 31
>2870 c9 01 d0 10 ad 11 ff 09
>2878 11 8d 11 ff ad 12 ff 09
>2880 01 8d 12 ff ad 04 31 c9
>2888 01 d0 10 ad 11 ff 09 12
>2890 8d 11 ff ad 12 ff 09 01
>2898 8d 12 ff a0 03 a9 05 ae
>28a0 06 31 e0 01 d0 03 8d 06
>28a8 31 ae 07 31 e0 01 d0 03
>28b0 8d 07 31 ae 08 31 e0 01
>28b8 d0 03 8d 08 31 ae 09 31
>28c0 e0 01 d0 03 8d 09 31 ae
>28c8 0a 31 e0 01 d0 03 8d 0a
>28d0 31 ae 06 31 e0 00 f0 0c
>28d8 ad 11 ff 09 4a ea 8d 11
>28e0 ff ce 06 31 ae 07 31 e0
>28e8 00 f0 0c ad 11 ff 09 4a
>28f0 ea 8d 11 ff ce 07 31 ae
>28f8 08 31 e0 00 f0 0b ad 11
>2900 ff 09 4b 8d 11 ff ce 08
>2908 31 ae 09 31 e0 00 f0 0b
>2910 ad 11 ff 09 4a 8d 11 ff
>2918 ce 09 31 ae 0a 31 e0 00
>2920 f0 0b ad 11 ff 09 4c 8d
>2928 11 ff ce 0a 31 60 ad 00
>2930 31 c9 01 d0 28 ae 12 31
>2938 e0 01 f0 21 ae 00 ff e0
>2940 01 f0 0b e0 05 f0 07 e0
>2948 aa d0 03 4c 5d 29 a2 01
>2950 8e 12 31 a6 da 8e 13 31
>2958 a6 db 8e 14 31 60 ad 01
>2960 31 c9 01 d0 28 ae 15 31
>2968 e0 01 f0 21 ae 00 ff e0
>2970 01 f0 0b e0 05 f0 07 e0
>2978 aa d0 03 4c 5d 29 a2 01
>2980 8e 15 31 a6 dc 8e 16 31
>2988 a6 dd 8e 17 31 60 ad 02
>2990 31 c9 01 d0 28 ae 18 31
>2998 e0 01 f0 21 ae 00 ff e0
>29a0 01 f0 0b e0 05 f0 07 e0
>29a8 aa d0 03 4c bd 29 a2 01
>29b0 8e 18 31 a6 de 8e 19 31
>29b8 a6 df 8e 1a 31 60 00 00
>29c0 00 00 00 00 00 00 00 00
>29c8 00 00 00 00 00 00 00 00
>29d0 ff ff ff ff ff ff ff ff
>29d8 ff ff ff ff ff ff ff ff
>29e0 00 00 00 00 00 00 00 00
>29e8 00 00 00 00 00 00 00 00
>29f0 ff ff ff ff ff ff ff ff
>29f8 ff ff ff ff ff ff ff ff
>2a00 ff ff ff ff ff ff ff ff
>2a08 ff ff ff ff ff ff ff ff
>2a10 00 00 00 00 00 00 00 00
>2a18 00 00 00 00 00 00 00 00
>2a20 ff ff ff ff ff ff ff ff
>2a28 ff ff ff ff ff ff ff ff
>2a30 00 00 00 00 00 00 00 00
>2a38 00 00 00 00 00 00 00 00
>2a40 00 00 00 00 00 00 00 00
>2a48 00 00 00 00 00 00 00 00
>2a50 ff ff ff ff ff ff ff ff
>2a58 ff ff ff ff ff ff ff ff
>2a60 00 00 00 00 00 00 00 00
>2a68 00 00 00 00 00 00 00 00
>2a70 ff ff ff ff ff ff ff ff
>2a78 ff ff ff ff ff ff ff ff
>2a80 ff ff ff ff ff ff ff ff
>2a88 ff ff ff ff ff ff ff ff
```

```
>2a90 00 00 00 00 00 00 00 00
>2a98 00 00 00 00 00 00 00 00
>2aa0 ff ff ff ff ff ff ff ff
>2aa8 ff ff ff ff ff ff ff ff
>2ab0 00 00 00 00 00 00 00 00
>2ab8 00 00 00 00 00 00 00 00
>2ac0 00 00 00 00 00 00 00 00
>2ac8 00 00 00 00 00 00 00 00
>2ad0 ff ff ff ff ff ff ff ff
>2ad8 ff ff ff ff ff ff ff ff
>2ae0 00 00 00 00 00 00 00 00
>2ae8 00 00 00 00 00 00 00 00
>2af0 ff ff ff ff ff ff ff ff
>2af8 ff ff ff ff ff ff ff ff
>2b00 00 00 00 00 00 00 00 00
>2b08 00 00 00 00 00 00 00 00
>2b10 ff ff ff ff ff ff ff ff
>2b18 ff ff ff ff ff ff ff ff
>2b20 00 00 00 00 00 00 00 00
>2b28 00 00 00 00 00 00 00 00
>2b30 ff ff ff ff ff ff ff ff
>2b38 ff ff ff ff ff ff ff ff
>2b40 ff ff ff ff ff ff ff ff
>2b48 ff ff ff ff ff ff ff ff
>2b50 00 00 00 00 00 00 00 00
>2b58 00 00 00 00 00 00 00 00
>2b60 ff ff ff ff ff ff ff ff
>2b68 ff ff ff ff ff ff ff ff
>2b70 00 00 00 00 00 00 00 00
>2b78 00 00 00 00 00 00 00 00
>2b80 00 00 00 00 00 00 00 00
>2b88 00 00 00 00 00 00 00 00
>2b90 ff ff ff ff ff ff ff ff
>2b98 ff ff ff ff ff ff ff ff
>2ba0 00 00 00 00 00 00 00 00
>2ba8 00 00 00 00 00 00 00 00
>2bb0 ff ff ff ff ff ff ff ff
>2bb8 ff ff ff ff ff ff ff ff
>2bc0 ff ff ff ff ff ff ff ff
>2bc8 ff ff ff ff ff ff ff ff
>2bd0 00 00 00 00 00 00 00 00
>2bd8 00 00 00 00 00 00 00 00
>2be0 ff ff ff ff ff ff ff ff
>2be8 ff ff ff ff ff ff ff ff
>2bf0 00 00 00 00 00 00 00 00
>2bf8 00 00 00 00 00 00 00 00
>2c00 00 00 00 00 00 00 00 00
>2c08 00 00 00 00 00 00 00 00
>2c10 ff ff ff ff ff ff ff ff
>2c18 ff ff ff ff ff ff ff ff
>2c20 00 00 00 00 00 00 00 00
>2c28 00 00 00 00 00 00 00 00
>2c30 ff ff ff ff ff ff ff ff
>2c38 ff ff ff ff ff ff ff ff
>2c40 ff ff ff ff ff ff ff ff
>2c48 ff ff ff ff ff ff ff ff
>2c50 00 00 00 00 00 00 00 00
>2c58 00 00 00 00 00 00 00 00
>2c60 ff ff ff ff ff ff ff ff
>2c68 ff ff ff ff ff ff ff ff
>2c70 00 00 00 00 00 00 00 00
>2c78 00 00 00 00 00 00 00 00
>2c80 00 00 00 00 00 00 00 00
>2c88 00 00 00 00 00 00 00 00
>2c90 ff ff ff ff ff ff ff ff
>2c98 ff ff ff ff ff ff ff ff
>2ca0 00 00 00 00 00 00 00 00
>2ca8 00 00 00 00 00 00 00 00
>2cb0 ff ff ff ff ff ff ff ff
>2cb8 ff ff ff ff ff ff ff ff
>2cc0 ff ff ff ff ff ff ff ff
>2cc8 ff ff ff ff ff ff ff ff
>2cd0 00 00 00 00 00 00 00 00
>2cd8 00 00 00 00 00 00 00 00
>2ce0 ff ff ff ff ff ff ff ff
>2ce8 ff ff ff ff ff ff ff ff
>2cf0 00 00 00 00 00 00 00 00
>2cf8 00 00 00 00 00 00 00 00
>2d00 00 00 00 00 00 00 00 00
>2d08 00 00 00 00 00 00 00 00
>2d10 ff ff ff ff ff ff ff ff
>2d18 ff ff ff ff ff ff ff ff
>2d20 00 00 00 00 00 00 00 00
>2d28 00 00 00 00 00 00 00 00
>2d30 ff ff ff ff ff ff ff ff
>2d38 ff ff ff ff ff ff ff ff
```

```
>2d40 ff ff ff ff ff ff ff ff
>2d48 ff ff ff ff ff ff ff ff
>2d50 00 00 00 00 00 00 00 00
>2d58 00 00 00 00 00 00 00 00
>2d60 ff ff ff ff ff ff ff ff
>2d68 ff ff ff ff ff ff ff ff
>2d70 00 00 00 00 00 00 00 00
>2d78 00 00 00 00 00 00 00 00
>2d80 00 00 00 00 00 00 00 00
>2d88 00 00 00 00 00 00 00 00
>2d90 ff ff ff ff ff ff ff ff
>2d98 ff ff ff ff ff ff ff ff
>2da0 00 00 00 00 00 00 00 00
>2da8 00 00 00 00 00 00 00 00
>2db0 ff ff ff ff ff ff ff ff
>2db8 ff ff ff ff ff ff ff ff
>2dc0 ff ff ff ff ff ff ff ff
>2dc8 ff ff ff ff ff ff ff ff
>2dd0 00 00 00 00 00 00 00 00
>2dd8 00 00 00 00 00 00 00 00
>2de0 ff ff ff ff ff ff ff ff
>2de8 ff ff ff ff ff ff ff ff
>2df0 00 00 00 00 00 00 00 00
>2df8 00 00 00 00 00 00 00 00
>2e00 00 00 00 00 00 00 00 00
>2e08 00 00 00 00 00 00 00 00
>2e10 ff ff ff ff ff ff ff ff
>2e18 ff ff ff ff ff ff ff ff
>2e20 00 00 00 00 00 00 00 00
>2e28 00 00 00 00 00 00 00 00
>2e30 ff ff ff ff ff ff ff ff
>2e38 ff ff ff ff ff ff ff ff
>2e40 ff ff ff ff ff ff ff ff
>2e48 ff ff ff ff ff ff ff ff
>2e50 00 00 00 00 00 00 00 00
>2e58 00 00 00 00 00 00 00 00
>2e60 ff ff ff ff ff ff ff ff
>2e68 ff ff ff ff ff ff ff ff
>2e70 00 00 00 00 00 00 00 00
>2e78 00 00 00 00 00 00 00 00
>2e80 00 00 00 00 00 00 00 00
>2e88 00 00 00 00 00 00 00 00
>2e90 ff ff ff ff ff ff ff ff
>2e98 ff ff ff ff ff ff ff ff
>2ea0 00 00 00 00 00 00 00 00
>2ea8 00 00 00 00 00 00 00 00
>2eb0 ff ff ff ff ff ff ff ff
>2eb8 ff ff ff ff ff ff ff ff
>2ec0 ff ff ff ff ff ff ff ff
>2ec8 ff ff ff ff ff ff ff ff
>2ed0 00 00 00 00 00 00 00 00
>2ed8 00 00 00 00 00 00 00 00
>2ee0 ff ff ff ff ff ff ff ff
>2ee8 ff ff ff ff ff ff ff ff
>2ef0 00 00 00 00 00 00 00 00
>2ef8 00 00 00 00 00 00 00 00
>2f00 00 00 00 00 00 00 00 00
>2f08 00 00 00 00 00 00 00 00
>2f10 ff ff ff ff ff ff ff ff
>2f18 ff ff ff ff ff ff ff ff
>2f20 00 00 00 00 00 00 00 00
>2f28 00 00 00 00 00 00 00 00
>2f30 ff ff ff ff ff ff ff ff
>2f38 ff ff ff ff ff ff ff ff
>2f40 ff ff ff ff ff ff ff ff
>2f48 ff ff ff ff ff ff ff ff
>2f50 00 00 00 00 00 00 00 00
>2f58 00 00 00 00 00 00 00 00
>2f60 ff ff ff ff ff ff ff ff
>2f68 ff ff ff ff ff ff ff ff
>2f70 00 00 00 00 00 00 00 00
>2f78 00 00 00 00 00 00 00 00
>2f80 00 00 00 00 00 00 00 00
>2f88 00 00 00 00 00 00 00 00
>2f90 ff ff ff ff ff ff ff ff
>2f98 ff ff ff ff ff ff ff ff
>2fa0 00 00 00 00 00 00 00 00
>2fa8 00 00 00 00 00 00 00 00
>2fb0 ff ff ff ff ff ff ff ff
>2fb8 ff ff ff ff ff ff ff ff
>2fc0 ff ff ff ff ff ff ff ff
>2fc8 ff ff ff ff ff ff ff ff
>2fd0 00 00 00 00 00 00 00 00
>2fd8 00 00 00 00 00 00 00 00
>2fe0 ff ff ff ff ff ff ff ff
>2fe8 ff ff ff ff ff ff ff ff
```

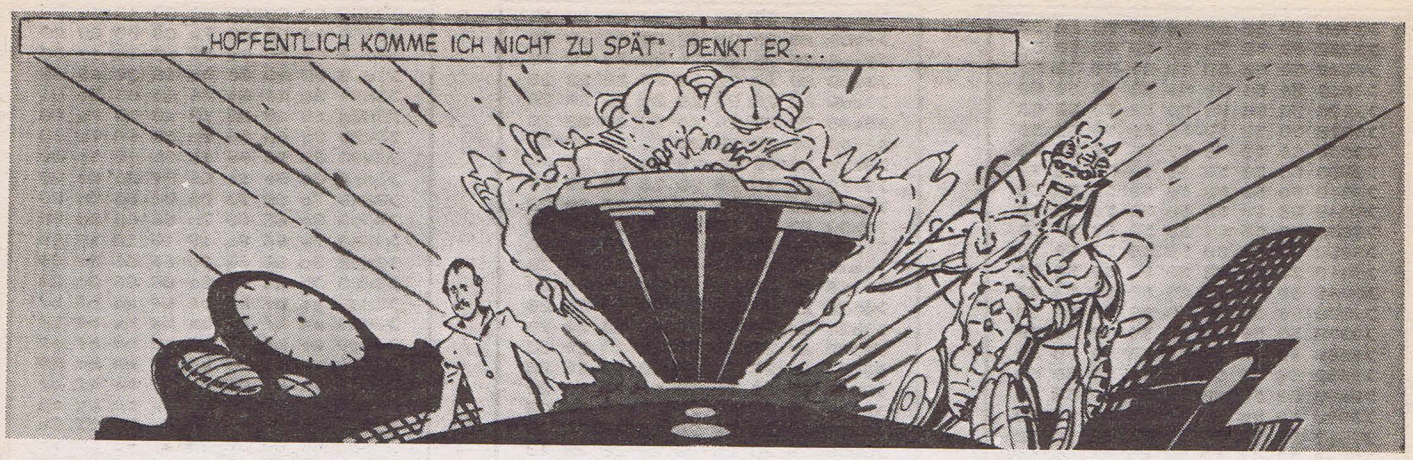
```
>2ff0 00 00 00 00 00 00 00 00
>2ff8 00 00 00 00 00 00 00 00
>3000 ae 12 31 e0 01 d0 49 ae
>3008 13 31 86 e6 ae 14 31 86
>3010 e7 a2 00 a9 20 81 e6 a5
>3018 e6 18 69 28 85 e6 a5 e7
>3020 69 00 85 e7 a1 e6 c9 20
>3028 d0 11 a9 4b 81 e6 a5 e6
>3030 8d 13 31 a5 e7 8d 14 31
>3038 4c b5 30 c9 40 d0 03 4c
>3040 00 32 c9 41 d0 03 4c 00
>3048 32 c9 42 d0 03 4c 00 32
>3050 c9 0a f0 01 60 a2 00 8e
>3058 12 31 60 ae 15 31 e0 01
>3060 d0 49 ae 16 31 86 e6 ae
>3068 17 31 86 e7 a2 00 a9 20
>3070 81 e6 a5 e6 18 69 28 85
>3078 e6 a5 e7 69 00 85 e7 a1
>3080 e6 c9 20 d0 11 a9 4b 81
>3088 e6 a5 e6 8d 16 31 a5 e7
>3090 8d 17 31 4c b5 30 c9 40
>3098 d0 03 4c 00 32 c9 41 d0
>30a0 03 4c 00 32 c9 42 d0 03
>30a8 4c 00 32 c9 0a f0 01 60
>30b0 a2 00 8e 15 31 60 00 00
>30b8 00 00 00 00 00 00 00 00
>30c0 00 00 00 00 00 00 00 00
>30c8 00 00 00 00 00 00 00 00
>30d0 ff ff ff ff ff ff ff ff
>30d8 ff ff ff ff ff ff ff ff
>30e0 00 00 00 00 00 00 00 00
>30e8 00 00 00 00 00 00 00 00
>30f0 ff ff ff ff ff ff ff ff
>30f8 ff ff ff ff ff ff ff ff
>3100 01 00 00 00 00 00 00 00
>3108 00 00 00 03 02 00 00 00
>3110 00 00 00 b8 0d 00 65 0f
>3118 00 21 0f 00 5c 0f 01 b9
>3120 0d ff ff ff ff ff ff ff
>3128 ff ff ff ff ff ff ff ff
>3130 00 00 00 00 00 00 00 00
>3138 00 00 00 00 00 00 00 00
>3140 00 00 00 00 00 00 00 00
>3148 00 00 a2 00 a0 00 20 f0
>3150 ff a9 99 20 d2 ff ea ea
>3158 ea ea a2 3f 8e 11 ff a9
>3160 3c 8d 13 ff a9 c0 8d 12
>3168 ff a9 18 8d 07 ff a9 80
>3170 8d 47 05 a9 06 8d 15 ff
>3178 ea a9 51 8d 16 ff a9 42
>3180 8d 17 ff a9 99 20 d2 ff
>3188 a2 00 8e 19 ff ea a0 31
>3190 ae 0c 31 ae a7 0f ae 0b
>3198 31 ae a8 0f 4c a0 31 ea
>31a0 a9 00 85 d8 85 da a9 34
>31a8 85 d9 a9 0c 85 db a9 01
>31b0 a2 04 a0 00 b1 d8 91 da
>31b8 88 d0 f9 e6 d9 e6 db ca
>31c0 d0 f2 20 40 32 a9 00 85
>31c8 d8 a9 08 85 d9 a9 6d a0
>31d0 00 a2 04 91 d8 88 d0 fb
>31d8 e6 d9 ca d0 f6 60 00 8e
>31e0 86 0f 60 00 00 00 00 00
>31e8 00 00 00 00 00 00 00 00
>31f0 00 00 00 00 00 00 31 8e
>31f8 86 0f 60 ff ff ff ff ff
>3200 a9 4f a2 00 81 d8 e6 d8
>3208 81 d8 e6 d8 81 d8 a0 00
>3210 c0 ff f0 0d c8 ae 00 ff
>3218 8e 19 ff 8e 19 ff 4c 12
>3220 32 a2 00 8e 19 ff 20 80
>3228 33 a2 00 8e 05 31 20 cb
>3230 23 4c 00 33 00 00 00 00
>3238 00 00 00 00 00 00 00 00
>3240 a2 00 8e 86 0f 60 ae 10
>3248 31 8e 85 0f ae 0f 31 8e
>3250 86 0f ae 0e 31 8e 87 0f
>3258 ae 0d 31 8e 88 0f ae 0c
>3260 31 8e 89 0f ae 0b 31 8e
>3268 8a 0f a2 00 8e 8b 0f 8e
>3270 8c 0f 60 ae 1e 31 e0 01
>3278 d0 49 ae 1f 31 86 e6 ae
>3280 20 31 86 e7 a2 00 a9 20
>3288 81 e6 a5 e6 18 69 28 85
>3290 e6 a5 e7 69 00 85 e7 a1
>3298 e6 c9 20 d0 11 a9 4b 81
>32a0 e6 a5 e6 8d 1f 31 a5 e7
>32a8 8d 20 31 4c b2 24 c9 40
>32b0 d0 03 4c 00 32 c9 41 d0
>32b8 03 4c 00 32 c9 42 d0 03
>32c0 4c 00 32 c9 0a f0 01 60
>32c8 a2 00 8e 1e 31 60 ad 04
>32d0 31 c9 01 d0 28 ae 1e 31
>32d8 e0 01 f0 21 ae 00 ff e0
>32e0 01 f0 0b e0 05 f0 07 e0
>32e8 aa d0 03 4c e2 24 a2 01
>32f0 8e 1e 31 a6 e2 8e 1f 31
>32f8 a6 e3 8e 20 31 60 ff ff
>3300 20 5a 31 ae 11 31 8e 82
>3308 0f ae 10 31 8e 83 0f ae
>3310 0f 31 8e 84 0f ae 0e 31
>3318 8e 85 0f ae 0d 31 8e 86
>3320 0f ae 0c 31 8e 87 0f ae
>3328 0b 31 8e 88 0f a2 00 8e
>3330 89 0f 8e 8a 0f 20 a1 25
>3338 20 b3 25 a9 00 85 d8 85
>3340 da a9 38 85 d9 a9 0c 85
>3348 db a9 01 a2 04 a0 00 b1
>3350 d8 91 da 88 d0 f9 e6 d9
>3358 e6 db ca d0 f2 20 c5 31
>3360 4c 63 33 20 98 25 a2 00
>3368 8e 1b 31 8e 1e 31 4c 00
>3370 24 00 00 00 00 00 00 00
>3378 00 00 00 00 00 00 00 00
>3380 a2 00 a9 00 a0 00 e8 e0
>3388 05 d0 fb c8 c0 20 d0 f6
>3390 60 ff ff ff ff ff ff ff
>3398 ff ff ff ff ff ff ff ff
>33a0 00 00 00 00 00 00 00 00
>33a8 00 00 00 00 00 00 00 00
>33b0 ff ff ff ff ff ff ff ff
>33b8 ff ff ff ff ff ff ff ff
>33c0 ff ff ff ff ff ff ff ff
>33c8 ff ff ff ff ff ff ff ff
>33d0 00 00 00 00 00 00 00 00
>33d8 00 00 00 00 00 00 00 00
>33e0 ff ff ff ff ff ff ff ff
>33e8 ff ff ff ff ff ff ff ff
>33f0 00 00 00 00 00 00 00 00
>33f8 00 00 00 00 00 00 00 00
>3400 50 50 50 50 50 50 50 50
>3408 50 50 50 50 50 50 50 50
>3410 50 50 50 50 50 50 50 50
>3418 50 50 50 50 50 50 50 50
>3420 50 50 50 50 50 50 50 50
>3428 50 50 50 50 50 50 50 50
>3430 50 50 50 50 55 55 55 55
>3438 55 55 55 55 55 55 55 55
>3440 55 55 50 50 50 50 50 50
>3448 50 50 50 50 50 50 50 50
>3450 50 50 50 50 50 50 50 50
>3458 50 50 50 50 55 56 58 56
>3460 5a 5c 5e 60 62 56 5a 64
>3468 66 55 50 50 50 50 50 50
>3470 50 50 50 50 50 50 50 50
>3478 50 50 50 50 50 50 50 50
>3480 50 50 50 50 55 57 59 57
>3488 5b 5d 5f 61 63 57 5b 65
>3490 67 55 50 50 50 50 50 50
>3498 50 50 50 50 50 50 50 50
>34a0 50 50 50 50 50 50 50 50
>34a8 50 50 50 50 55 55 55 55
>34b0 55 55 55 55 55 55 55 55
>34b8 55 55 50 50 50 50 50 50
>34c0 50 50 50 50 50 50 50 50
>34c8 50 50 50 50 50 50 50 50
>34d0 50 50 50 50 50 50 50 50
>34d8 50 50 50 50 50 50 50 50
>34e0 50 50 50 50 50 50 50 50
>34e8 50 50 50 50 50 50 50 50
>34f0 51 52 53 54 52 53 52 51
>34f8 53 54 52 53 52 51 53 52
>3500 53 54 52 53 51 53 52 54
>3508 52 53 51 53 51 52 51 52
>3510 54 52 53 51 52 53 54 52
>3518 20 20 20 20 20 20 20 20
>3520 20 20 20 20 20 20 20 20
>3528 20 20 20 20 20 20 20 20
>3530 20 20 20 20 20 20 20 20
>3538 20 20 20 20 20 20 20 20
>3540 20 20 20 20 20 20 20 20
>3548 20 20 20 20 20 20 20 20
>3550 20 20 20 20 20 20 20 20
>3558 20 20 20 20 20 20 20 20
>3560 20 20 20 20 20 20 20 20
>3568 20 20 20 20 20 20 20 20
>3570 20 20 20 20 20 20 20 20
>3578 20 20 20 20 20 20 20 20
>3580 20 20 20 20 20 20 20 20
>3588 20 20 20 20 20 20 20 20
>3590 20 20 20 20 20 20 20 20
>3598 20 20 20 20 20 20 20 20
>35a0 20 20 20 20 20 20 20 20
>35a8 20 20 20 20 20 20 20 20
>35b0 20 20 20 20 20 20 20 20
>35b8 20 20 20 20 20 20 20 20
>35c0 20 20 20 20 20 20 20 20
>35c8 20 20 20 20 20 20 20 20
>35d0 20 20 20 20 20 20 20 20
>35d8 20 20 20 20 20 20 20 20
>35e0 20 20 20 20 20 20 20 20
>35e8 20 20 20 20 20 20 20 20
>35f0 20 20 20 20 20 20 20 20
>35f8 20 20 20 20 20 20 20 20
>3600 20 20 20 20 20 20 20 20
>3608 20 20 20 20 20 20 20 20
>3610 20 20 20 20 20 20 20 20
>3618 20 20 20 20 20 20 20 20
>3620 20 20 20 20 20 20 20 20
>3628 20 20 20 20 20 20 20 20
>3630 20 20 20 20 20 20 20 20
>3638 20 20 20 20 20 20 20 20
>3640 20 20 20 20 20 20 20 20
>3648 20 20 20 20 20 20 20 20
>3650 20 20 20 20 20 20 20 20
>3658 20 20 20 20 20 20 20 20
>3660 20 20 20 20 20 20 20 20
>3668 20 20 20 20 20 20 20 20
>3670 20 20 20 20 20 20 20 20
>3678 20 20 20 20 20 20 20 20
>3680 20 20 20 20 20 20 20 20
>3688 20 20 20 20 20 20 20 20
>3690 20 20 20 20 20 20 20 20
>3698 20 20 20 20 20 20 20 20
>36a0 20 20 20 20 20 20 20 20
>36a8 20 20 20 20 20 20 20 20
>36b0 20 20 20 20 20 20 20 20
>36b8 20 20 20 20 20 20 20 20
>36c0 20 20 20 20 20 20 20 20
>36c8 20 20 20 20 20 20 20 20
>36d0 20 20 20 20 20 20 20 20
>36d8 20 20 0b 10 0b 0c 0d 0c
>36e0 0e 0e 0f 0e 11 0e 0c 13
>36e8 12 20 20 20 20 20 20 20
>36f0 20 20 20 20 20 20 20 20
>36f8 20 20 20 20 20 20 20 20
>3700 20 20 20 20 20 20 20 20
>3708 20 20 20 20 20 20 20 20
>3710 20 20 20 20 20 20 20 20
>3718 20 20 20 20 20 20 20 20
>3720 20 20 20 20 20 20 20 14
>3728 16 17 0e 0e 20 0e 14 15
>3730 0b 17 20 12 0c 20 0e 12
>3738 15 16 12 20 20 20 20 20
>3740 20 20 20 20 20 20 20 20
>3748 20 20 20 20 20 20 20 20
>3750 20 20 20 20 20 20 20 20
>3758 20 20 20 20 20 20 20 20
>3760 20 20 20 20 20 20 20 20
>3768 20 20 20 20 20 20 20 20
>3770 20 20 20 20 20 20 20 0d
>3778 15 0e 12 20 0e 0b 0c 16
>3780 17 20 20 20 20 20 20 20
>3788 20 20 20 20 20 20 20 20
>3790 20 20 20 20 20 20 20 20
>3798 20 20 20 20 20 20 20 20
>37a0 20 20 20 20 20 20 20 20
>37a8 20 20 20 20 20 20 20 20
>37b0 20 20 20 20 20 20 20 20
>37b8 20 20 20 20 20 20 20 20
>37c0 20 20 20 20 20 20 20 20
>37c8 20 20 20 20 20 20 20 20
>37d0 20 20 20 20 20 20 20 20
>37d8 20 20 20 20 20 20 20 20
>37e0 20 20 20 20 20 20 20 20
>37e8 00 00 00 00 00 00 00 00
>37f0 ff ff ff ff ff ff ff ff
>37f8 ff ff ff ff ff ff ff ff
```

```
>3800 50 50 50 50 50 50 50 50
>3808 50 50 50 50 50 50 50 50
>3810 50 50 50 50 50 50 50 50
>3818 50 50 50 50 50 50 50 50
>3820 50 50 50 50 50 50 50 50
>3828 50 50 50 50 50 50 50 50
>3830 50 50 50 50 50 50 50 50
>3838 50 50 50 50 50 50 50 50
>3840 50 50 50 50 50 50 50 50
>3848 50 50 50 50 50 50 50 50
>3850 50 50 50 50 50 50 50 50
>3858 50 50 50 50 50 50 50 50
>3860 50 50 50 50 50 50 50 50
>3868 50 50 50 50 50 50 50 50
>3870 50 50 50 50 50 50 50 50
>3878 50 50 50 50 50 50 50 50
>3880 50 50 50 50 50 50 50 50
>3888 50 50 50 50 50 50 50 50
>3890 50 50 50 50 50 50 50 50
>3898 50 50 50 50 50 50 50 50
>38a0 51 52 54 53 51 52 54 52
>38a8 53 51 52 54 53 51 52 53
>38b0 52 54 53 52 51 53 52 54
>38b8 52 53 51 52 54 54 53 51
>38c0 54 52 54 52 53 54 52 53
>38c8 20 20 20 20 20 20 20 20
>38d0 20 20 20 20 20 20 20 20
>38d8 20 20 20 20 20 20 20 20
>38e0 20 20 20 20 20 20 20 20
>38e8 20 20 20 20 20 20 20 20
>38f0 20 20 20 20 20 20 20 20
>38f8 20 20 20 20 20 20 20 20
>3900 20 20 20 20 20 20 20 20
>3908 20 20 20 20 20 20 20 20
>3910 20 20 20 20 20 20 20 20
>3918 20 20 20 20 20 20 20 20
>3920 20 20 20 20 20 20 20 20
>3928 20 20 20 20 20 20 20 20
>3930 20 20 20 20 20 20 20 20
>3938 20 20 20 20 20 20 20 20
>3940 20 20 20 20 20 20 20 20
>3948 20 20 20 20 20 20 20 20
>3950 20 20 20 20 20 20 20 20
>3958 20 20 20 20 20 20 20 20
>3960 20 20 20 20 20 20 20 20
>3968 20 20 20 20 20 20 20 20
>3970 20 20 20 20 20 20 20 20
>3978 20 20 20 20 20 20 20 20
>3980 20 20 20 20 20 20 20 20
>3988 20 20 20 20 20 20 20 20
>3990 20 20 20 20 20 20 20 20
>3998 20 20 20 20 20 20 20 20
>39a0 20 20 20 20 20 20 20 20
>39a8 20 20 20 20 20 20 20 20
>39b0 20 20 20 20 20 20 20 20
>39b8 20 20 20 20 20 20 20 20
>39c0 20 20 20 20 20 20 20 20
>39c8 20 20 20 20 20 20 20 20
>39d0 20 20 20 20 20 20 20 20
>39d8 20 20 20 20 20 20 20 20
>39e0 20 20 20 20 20 20 20 20
>39e8 20 20 20 20 20 20 20 20
>39f0 20 20 20 20 20 20 20 20
>39f8 20 20 20 20 20 20 20 20
>3a00 20 20 20 20 20 20 20 20
>3a08 20 20 20 20 20 20 20 20
>3a10 20 20 20 20 20 20 20 20
>3a18 20 20 20 20 20 20 20 20
>3a20 20 20 20 20 20 20 20 20
>3a28 20 20 20 20 20 20 20 20
>3a30 20 20 20 20 20 20 20 20
>3a38 20 20 20 20 20 20 20 20
>3a40 20 20 20 20 20 20 20 20
>3a48 20 20 20 20 20 20 20 20
>3a50 20 20 20 20 20 20 20 20
>3a58 20 20 20 20 20 20 20 20
>3a60 20 20 20 20 20 20 20 20
>3a68 20 20 20 20 20 20 20 20
>3a70 20 20 20 20 20 20 20 20
>3a78 20 20 20 20 20 20 20 20
>3a80 20 20 20 20 20 20 20 20
>3a88 20 20 20 20 20 20 20 20
>3a90 20 20 20 20 20 20 20 20
>3a98 20 20 20 20 20 20 20 20
>3aa0 20 20 20 20 20 20 20 20
>3aa8 20 20 20 20 20 20 20 20
```

```
>3ab0 20 20 20 20 20 20 20 20
>3ab8 20 20 20 20 20 20 20 20
>3ac0 20 20 20 20 20 20 20 20
>3ac8 20 20 20 20 20 20 20 20
>3ad0 20 20 20 20 20 20 20 20
>3ad8 20 20 20 20 20 20 20 20
>3ae0 20 20 20 20 20 20 20 20
>3ae8 20 20 20 20 20 20 20 20
>3af0 20 20 20 20 20 20 20 20
>3af8 20 20 20 20 20 20 20 20
>3b00 20 20 20 20 20 20 20 20
>3b08 20 20 20 20 20 20 20 20
>3b10 20 20 20 20 20 20 20 20
>3b18 20 20 20 20 20 20 20 20
>3b20 20 20 20 20 20 20 20 20
>3b28 20 20 20 20 20 20 20 20
>3b30 20 20 20 20 20 20 20 20
>3b38 20 20 20 20 20 20 20 20
>3b40 20 20 20 20 20 20 20 20
>3b48 20 20 20 20 20 20 20 20
>3b50 20 20 20 20 20 20 20 20
>3b58 20 20 20 20 20 20 20 20
>3b60 20 20 20 20 20 20 20 20
>3b68 20 20 20 20 20 20 20 20
>3b70 0a 0a 0a 0a 0a 0a 0a 0a
>3b78 0a 0a 0a 0a 0a 0a 0a 0a
>3b80 0a 0a 0a 0a 0a 0a 0a 0a
>3b88 0a 0a 0a 0a 0a 0a 0a 0a
>3b98 0a 0a 0a 0a 0a 0a 0a 0a
>3b98 56 58 56 5a 5c 5e 60 62
>3ba0 56 5a 64 66 55 55 55 3a
>3ba8 3b 3c 3d 3e 3f 00 00 00
>3bb0 00 00 00 00 00 00 55 55
>3bb8 55 55 55 55 55 55 55 55

>3bc0 57 59 57 5b 5d 5f 61 63
>3bc8 57 5b 65 67 55 55 55 55
>3bd0 55 55 55 55 55 55 55 55
>3bd8 55 55 55 55 55 55 55 55
>3be0 55 55 55 55 55 55 55 55
>3be8 ff ff ff ff ff ff ff ff
>3bf0 00 00 00 00 00 00 00 00
>3bf8 00 00 00 00 00 00 00 00
>3c00 aa fe ee ee ee ee ee fe
>3c08 aa ae ae ae ae ae ae ae
>3c10 aa fe ae ae fe ea ea fe
>3c18 aa fe ae ae fe ae ae fe
>3c20 aa ea ee ee fe ae ae ae
>3c28 aa fe ea ea fe ae ae fe
>3c30 aa fe ea ea fe ee ee fe
>3c38 aa fe ae ae be ae ae ae
>3c40 aa fe ee ee fe ee ee fe
>3c48 aa fe ee ee fe ae ae fe
>3c50 00 00 00 00 aa aa aa aa
>3c58 fc fc c0 c0 c0 c0 fc fc
>3c60 fc fc cc cc cc cc fc fc
>3c68 c0 c0 c0 c0 c0 c0 fc fc
>3c70 fc c0 c0 fc fc 0c 0c fc
>3c78 cc cc cc cc cc cc fc fc
>3c80 00 00 00 00 00 00 c0 c0
>3c88 00 00 00 3c 3c 00 00 00
>3c90 fc fc 30 30 30 30 30 30
>3c98 fc fc c0 c0 f0 f0 c0 c0
>3ca0 fc fc cc cc fc fc c0 c0
>3ca8 fc fc cc cc fc fc cc cc
>3cb0 fc fc cc cc fc fc f0 cc
>3cb8 fc fc c0 f0 f0 c0 fc fc
>3cc0 f0 f0 cc cc cc cc f0 f0
>3cc8 66 66 66 3c 18 18 18 00
>3cd0 7e 06 0c 18 30 60 7e 00
>3cd8 3c 30 30 30 30 30 3c 00
>3ce0 0c 12 30 7c 30 62 fc 00
>3ce8 3c 0c 0c 0c 0c 0c 3c 00
>3cf0 00 18 3c 7e 18 18 18 18
>3cf8 00 10 30 7f 7f 30 10 00
>3d00 00 00 00 00 00 00 00 00
>3d08 18 18 18 18 00 00 18 00
>3d10 66 66 66 00 00 00 00 00
>3d18 66 66 ff 66 ff 66 66 00
>3d20 18 3e 60 3c 06 7c 18 00
>3d28 62 66 0c 18 30 66 46 00
>3d30 3c 66 3c 38 67 66 3f 00
>3d38 06 0c 18 00 00 00 00 00
>3d40 0c 18 30 30 30 18 0c 00
>3d48 30 18 0c 0c 0c 18 30 00
>3d50 00 66 3c ff 3c 66 00 00
>3d58 00 18 18 7e 18 18 00 00
```

```
>3d60 00 00 00 00 00 18 18 30
>3d68 00 00 00 7e 00 00 00 00
>3d70 00 00 00 00 00 18 18 00
>3d78 00 03 06 0c 18 30 60 00
>3d80 3c 66 6e 76 66 66 3c 00
>3d88 18 18 38 18 18 18 7e 00
>3d90 3c 66 06 0c 30 60 7e 00
>3d98 3c 66 06 1c 06 66 3c 00
>3da0 06 0e 1e 66 7f 06 06 00
>3da8 7e 60 7c 06 06 66 3c 00
>3db0 3c 66 60 7c 66 66 3c 00
>3db8 7e 66 0c 18 18 18 18 00
>3dc0 3c 66 66 3c 66 66 3c 00
>3dc8 3c 66 66 3e 06 66 3c 00
>3dd0 aa bf bf ba bf ab bf bf
>3dd8 aa bf bf ba ba ba bf bf
>3de0 aa bf bf bd bd bd bf bf
>3de8 aa bf bd bd bf be bd bd
>3df0 aa bf bf ba bf ba bf bf
>3df8 aa aa be be aa be be aa
>3e00 00 00 00 00 2a bc 95 2a
>3e08 00 00 30 3c aa f3 55 aa
>3e10 00 00 00 00 a8 cc 54 a8
>3e18 00 00 00 00 00 fc 3f 0f
>3e20 00 00 10 11 15 aa ff ff
>3e28 00 00 00 00 04 a8 ff ff
>3e30 00 00 00 00 00 3f ff fc
>3e38 30 30 30 30 20 30 20 30
>3e40 00 00 00 00 00 a0 aa 0a
>3e48 00 00 00 00 10 ff aa aa
>3e50 00 00 04 44 44 fc aa aa
>3e58 00 00 00 00 04 2a a8 a0
>3e60 00 00 00 00 00 03 ea 3f
>3e68 00 00 00 00 00 c0 ab fc
>3e70 3c 14 ff d7 ff eb 14 3c
>3e78 04 a8 52 24 d9 7e 3c 1c
>3e80 ff ff ff ff ff ff ff ff
>3e88 ff ff ff fb e6 95 55 55
>3e90 ff ff ff ff bf 6e 59 55
>3e98 ff ff fb e6 91 55 55 55
>3ea0 ff ff ff bf 6e 59 55 55
>3ea8 aa aa aa aa aa aa aa aa
>3eb0 bf bf bf bf bc bc bc bc
>3eb8 bc bc bc bc bf bf bf bf
>3ec0 fc fc fc fc aa aa aa aa
>3ec8 aa aa aa aa fc fc fc fc
>3ed0 fc fc fc fc bc bc bc bc
>3ed8 bc bc bc bc fc fc fc fc
>3ee0 bc bc bc bc bf bf bf bf
>3ee8 bf bf bf bf bf bf bf bf
>3ef0 bc bc 3c 3c 3c 3c fc fc
>3ef8 fc fc 3c 3c 3c 3c 3c 3c
>3f00 bf bf bf bf bf bf bf bf
>3f08 bf bf bf bf af af ab ab
>3f10 3c 3c 3c 3c 3c 3c 3c 3c
>3f18 3c 3c 3c 3c f2 f2 ca ca
>3f20 bc bc bc bc af af af af
>3f28 00 ab ab ab af af bf bf
>3f30 bc bc bc bc 3c 3c 3c 3c
>3f38 fc fc f2 f2 ca ca 2a 2a
>3f40 aa 96 96 a6 a6 a6 a6 a6
>3f48 00 ff 00 ff 00 00 ff ff
>3f50 fc fc cc cc cc fc fc fc
>3f58 18 18 18 1f 1f 18 18 18
>3f60 00 00 00 00 0f 0f 0f 0f
>3f68 18 18 18 1f 1f 00 00 00
>3f70 00 00 00 f8 f8 18 18 18
>3f78 00 00 00 00 00 00 ff ff
>3f80 00 00 00 1f 1f 18 18 18
>3f88 18 18 18 ff ff 00 00 00
>3f90 00 00 00 ff ff 18 18 18
>3f98 18 18 18 f8 30 36 36 34
>3fa0 03 ff 30 30 2c 20 4f 4b
>3fa8 2c 30 30 2c 30 30 00 e0
>3fb0 07 07 07 07 07 07 07 07
>3fb8 ff ff 00 00 00 00 00 00
>3fc0 ff ff ff 00 00 00 00 57
>3fc8 45 52 28 ff 30 31 2c 20
>3fd0 46 49 4c 45 53 20 53 43
>3fd8 52 41 54 43 48 45 44 2c
>3fe0 30 31 2c 30 30 00 4a 4a
>3fe8 4a 4a 4a 4a 4a 4a 43 4f
>3ff0 4e 56 4f 59 0d ff 8d 3e
>3ff8 ff 4c a4 f2 f6 ff 0f 08
>4000 ff ff ff ff ff ff ff ff
```



»Los, los – Kohle raus«, brüllt RICH FREDDY

Bei diesem Spiel sind Sie ein Geldeintreiber, genannt Freddy. Aber da Freddy zu ängstlich ist, in jedem Haus um das Geld zu kämpfen, steuert er ein ferngesteuertes UFO mit dem er das Geld einsammelt. Ihre Aufgabe ist es nun das UFO so zu steuern, daß es nicht mit anderen Dingen, z. B. Spinnen zusammenstößt. Das UFO darf auch nicht beliebig oft die Mauern des Hauses berühren. Am unteren Bildschirmrand steht wie oft man noch die Mauer berühren darf. Ist diese Zahl null oder sind sie mit einem Wesen, das sie vom Geld abhalten soll, zusammengeprallt, so verlieren Sie eines Ihrer 5 Versuche in allen 4 Häusern das Geld einzusammeln.

Den Spielablauf können Sie mit SHIFT bzw. SHIFT LOCK anhalten. Zum nächsthöheren Level (nächstes Bild) kommen Sie, indem Sie RETURN drücken.

Teil 1

C-16 LISTING + CHECKSUMMEN (DC V1.0)

```
10 REM                                                          <153>
20 REM *** RICH FREDDY *** DATA PART ***                        <234>
30 REM                                                          <173>
40 REM                                                          <183>
50 SCNCLR:COLOR0,1:COLOR4,1:CHAR,13,10,"{YELLOW}
PLEASE{SPACE}WAIT...                                            <60>
60 POKE55,255:POKE56,55:CLR                                     <135>
70 FORL=15360TO16167:READT$:POKEL,DEC(T$):NEXT                  <82>
80 SYS16136                                                     <249>
90 FORL=14856TO15151:READT$:POKEL,DEC(T$):NEXT                  <60>
100 PRINT"{CLEAR BLACK DOWN2}L{SO}"CHR$(34)"RICH
{SPACE}FREDDY{SPACE}P.2"CHR$(34);                               <140>
110 PRINT",8":PRINT"{DOWN4}RUN{HOME}";:POKE1319,
13:POKE1320,13:POKE239,2                                        <69>
120 DATA AD,2B,3A,C9,9D,D0,0B,A9                                <56>
130 DATA B7,A2,76,8D,2B,3A,8E,33                                <110>
140 DATA 3A,60,A9,9D,A2,DE,4C,0B                                <159>
150 DATA 3C,A2,18,A0,12,18,20,F0                                <87>
160 DATA FF,A9,05,20,D2,FF,C6,D0                                <1>
170 DATA A6,D0,D0,09,A9,01,EA,EA                                <128>
180 DATA EA,EA,EA,85,D1,A9,00,20                                <244>
190 DATA 5F,A4,A9,20,4C,D2,FF,A2                                <151>
200 DATA 00,A0,00,EE,15,FF,E8,D0                                <154>
210 DATA FA,C8,D0,F7,60,86,E6,84                                <79>
220 DATA E7,A0,28,20,81,9A,20,91                                <94>
230 DATA A2,A4,E7,20,81,9A,20,7B                                <98>
240 DATA A0,20,91,A2,A4,E6,20,81                                <90>
250 DATA 9A,20,9E,9E,20,91,A2,A9                                <120>
260 DATA 0C,85,62,A9,00,85,63,A2                                <100>
270 DATA 90,3B,20,CE,A2,20,9E,9E                                <198>
280 DATA 20,E4,9D,84,E6,85,E7,A0                                <105>
290 DATA 00,B1,E6,60,85,E7,18,20                                <82>
300 DATA F0,FF,A5,E7,20,D2,FF,60                                <6>
310 DATA A9,20,A4,D4,A6,D5,20,8C                                <251>
320 DATA 3C,E6,D4,A6,D4,A4,D5,20                                <223>
330 DATA 4D,3C,C9,20,F0,16,C9,45                                <244>
340 DATA D0,07,A9,01,85,D1,4C,C4                                <200>
350 DATA 3C,A5,D2,85,D4,A5,D3,85                                <76>
360 DATA D5,EA,EA,EA,A9,48,8D,3B                                <12>
370 DATA 05,A9,6A,A4,D4,A6,D5,4C                                <230>
380 DATA 8C,3C,A2,00,B5,D6,85,D2                                <20>
390 DATA B5,D7,85,D3,B5,D8,85,D4                                <60>
400 DATA B5,D9,85,D5,86,E5,20,98                                <74>
410 DATA 3C,A5,D2,A6,E5,95,D6,A5                                <154>
420 DATA D3,95,D7,A5,D4,95,D8,A5                                <156>
430 DATA D5,95,D9,E8,E8,E8,E8,E0                                <82>
440 DATA 0C,D0,D1,60,AD,43,05,F0                                <95>
450 DATA 03,4C,04,3D,60,A9,20,A4                                <212>
460 DATA E2,A6,E3,20,8C,3C,C6,E3                                <177>
470 DATA A6,E2,A4,E3,20,4D,3C,C9                                <126>
480 DATA 20,F0,26,C9,45,D0,07,A9                                <198>
490 DATA 01,85,D1,4C,75,3D,C9,46                                <129>
500 DATA F0,F5,AD,16,3D,C9,C6,F0                                <146>
510 DATA 08,A9,C6,8D,16,3D,4C,16                                <104>
520 DATA 3D,A9,E6,8D,16,3D,4C,16                                <173>
530 DATA 3D,C6,E2,A6,E2,A4,E3,20                                <123>
540 DATA 4D,3C,C9,20,F0,1F,C9,45                                <231>
550 DATA F0,CD,C9,46,F0,C9,AD,49                                <244>
560 DATA 3D,C9,C6,F0,08,A9,C6,8D                                <189>
570 DATA 49,3D,4C,49,3D,A9,E6,8D                                <67>
580 DATA 49,3D,4C,49,3D,A9,6B,8D                                <29>
590 DATA 3B,05,A9,71,A4,E2,A6,E3                                <179>
600 DATA 4C,8C,3C,A9,00,8D,3B,05                                <100>
610 DATA A9,79,A4,E8,A6,E9,20,8C                                <11>
620 DATA 3C,C6,E9,A6,E8,A4,E9,20                                <233>
630 DATA 4D,3C,C9,59,F0,14,AD,91                                <92>
640 DATA 3D,C9,C6,F0,08,A9,C6,8D                                <13>
650 DATA 91,3D,4C,91,3D,A9,E6,4C                                <232>
660 DATA A7,3D,A9,78,A4,E8,A6,E9                                <173>
670 DATA 20,8C,3C,A5,E9,C5,F8,D0                                <149>
680 DATA 04,A5,FE,F0,01,60,A9,01                                <77>
690 DATA 85,FE,A6,E8,CA,86,F5,A4                                <240>
700 DATA E9,84,F6,4C,E5,3D,A9,20                                <171>
710 DATA A4,F5,A6,F6,20,8C,3C,C6                                <171>
720 DATA F5,A6,F5,A4,F6,20,4D,3C                                <38>
730 DATA C9,20,F0,10,C9,46,D0,07                                <130>
740 DATA A9,01,85,D1,4C,FC,3D,A9                                <5>
750 DATA 00,85,FE,60,A9,00,8D,3B                                <186>
760 DATA 05,A9,7A,A4,F5,A6,F6,4C                                <221>
770 DATA 8C,3C,A9,FB,8D,08,FF,AD                                <178>
780 DATA 08,FF,85,03,6A,90,0E,6A                                <152>
790 DATA 90,10,6A,90,12,6A,90,14                                <114>
```

```
800 DATA A5,04,4C,12,3E,C6,F8,4C                <25>
810 DATA 36,3E,E6,F8,4C,36,3E,C6                <87>
820 DATA F7,4C,36,3E,E6,F7,A6,F7                <120>
830 DATA A4,F8,20,4D,3C,C9,20,F0                <174>
840 DATA 66,EA,A2,08,DD,32,03,F0                <45>
850 DATA 1A,CA,D0,F8,C9,50,F0,21                <76>
860 DATA C9,56,F0,3B,C9,57,F0,49                <253>
870 DATA C9,45,F0,4B,C9,46,F0,47                <180>
880 DATA 4C,E4,3E,20,19,3C,A5,F9                <244>
890 DATA A6,FA,85,F7,86,F8,4C,DB                <138>
900 DATA 3E,A5,FC,18,69,01,85,FC                <228>
910 DATA A9,63,8D,3B,05,A0,09,A2                <242>
920 DATA 00,18,20,F0,FF,A9,00,A6                <81>
930 DATA FC,20,5F,A4,4C,A7,3E,A4                <228>
940 DATA F7,C8,A9,20,A6,F8,20,8C                <139>
950 DATA 3C,A5,FC,18,69,0A,4C,76                <117>
960 DATA 3E,A4,F7,88,4C,92,3E,A5                <73>
970 DATA FD,D0,07,E6,FD,E6,F7,4C                <206>
980 DATA 36,3E,C6,F7,A9,20,A4,F9                <200>
990 DATA A6,FA,20,8C,3C,20,D2,FF                <137>
1000 DATA A9,6D,8D,3B,05,A9,65,A4               <168>
1010 DATA F7,A6,F8,20,8C,3C,A9,66               <37>
1020 DATA 20,D2,FF,A5,F7,A6,F8,85               <193>
1030 DATA F9,86,FA,A5,03,85,04,A9               <198>
1040 DATA 00,85,FD,60,A9,01,85,D1               <155>
1050 DATA 4C,A7,3E,A2,00,A0,09,A9               <234>
1060 DATA FB,8D,08,FF,AD,08,FF,C9               <80>
1070 DATA FF,F0,04,C5,04,D0,06,CA               <193>
1080 DATA D0,ED,88,D0,EA,4C,0A,3E               <49>
1090 DATA A2,00,BD,00,D0,9D,00,38               <247>
1100 DATA BD,00,D1,9D,00,39,BD,00               <180>
1110 DATA D2,9D,00,3A,BD,00,D3,9D               <231>
1120 DATA 00,3B,CA,D0,E5,60,00,00               <234>
1130 REM ***********************                <37>
1140 DATA 00,01,02,C2,C0,C0,C2,32               <8>
1150 DATA 50,54,D4,A4,A0,50,A8,58               <5>
1160 DATA 36,15,16,12,00,00,00,01               <103>
1170 DATA 58,68,A8,A8,A0,50,50,50               <50>
1180 DATA 00,02,2A,9D,2A,00,01,00               <255>
1190 DATA 00,80,A8,DE,A8,00,40,00               <87>
1200 DATA 7E,18,7E,3C,18,00,00,00               <221>
1210 DATA 00,00,00,18,3C,7E,18,7E               <221>
1220 DATA 3C,AA,96,AA,24,18,24,18               <58>
1230 DATA 00,28,BE,28,00,00,00,00               <246>
1240 DATA 00,90,98,FC,FC,98,90,00               <196>
1250 DATA 00,09,19,3F,3F,19,09,00               <169>
1260 DATA 55,7F,7F,7F,55,7D,7D,7D               <75>
1270 DATA 55,DF,DF,DF,55,FD,FD,FD               <17>
1280 DATA 55,DD,DD,DD,55,FD,FD,FD               <230>
1290 DATA FF,E7,C1,97,C3,E9,83,E7               <55>
1300 DATA 00,28,96,AA,96,28,3C,FF               <222>
1310 DATA 00,18,05,1B,67,0C,13,64               <223>
1320 DATA 80,8C,D0,EC,F3,98,64,13               <30>
1330 DATA 80,80,80,80,80,80,80,80               <126>
1340 DATA 15,15,A9,A9,A9,A9,A9,A8               <44>
1350 DATA FF,EC,CB,AB,EB,EB,EC,FF               <162>
1360 DATA FF,E7,43,57,43,6B,C3,E7               <152>
1370 DATA 7B,19,3B,F9,FB,39,1B,79               <59>
1380 DATA 03,01,03,01,03,01,03,01               <92>
1390 DATA 00,00,00,6E,6E,00,00,00               <103>
1400 DATA 00,00,03,07,0E,1F,38,7C               <56>
1410 DATA E0,F0,70,38,39,DC,3F,7F               <21>
1420 DATA 01,03,07,F7,E3,00,FF,FF               <53>
1430 DATA FF,FE,80,F0,F8,78,F0,E0               <156>
1440 DATA 00,00,0F,09,09,0F,08,08               <121>
1450 DATA 00,00,7B,4A,4A,7B,52,4B               <52>
1460 DATA 00,00,3D,21,21,BD,05,FD               <214>
1470 DATA 00,00,93,1A,1E,D6,12,F2               <149>
1480 DATA 00,00,FF,48,48,4F,41,7F               <140>
1490 DATA 00,00,00,30,00,00,30,00               <91>
1500 DATA CC,CC,33,33,CC,CC,33,33               <120>
  ENDE DES LISTINGS
```

Teil 2

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```
10 REM"{SPACE2 SU S*19 SI                       <71>
20 REM"{SPACE2 S-}>>>{SPACE}RICH{SPACE}FREDDY{SP
ACE}<<<{S-                                      <41>
30 REM"{SPACE2 CQ S*19 CW                       <99>
40 REM"{SPACE2 S-}(C){SPACE}1986{SPACE}ANGEL-SOF
T{S-                                            <148>
50 REM"{SPACE2 S- SE19 S-                       <28>
60 REM"{SPACE2 S- SPACE4}WRITTEN{SPACE}BY{SPACE5
 S-                                             <211>
70 REM"{SPACE2 S-}-------------------{S-        <248>
80 REM"{SPACE2 S- SPACE2}OLIVER{SPACE2}DANGEL{SP
ACE3 S-                                         <209>
90 REM"{SPACE2 S- SPACE2}RIESBERGSTR.37{SPACE3 S
-                                               <70>
100 REM"{SPACE S- SPACE2}7157{SPACE}MURRHARDT{SP
ACE3 S-                                         <55>
110 REM"{SPACE SJ S*19 SK                       <81>
120 COLOR0,7,3:COLOR4,1:SCNCLR                  <34>
130 V=65280:POKEV+18,192:POKEV+19,56            <115>
140 POKEV+7,24:PRINTCHR$(8);                    <187>
150 POKEV+22,0:POKEV+23,119:POKEV+6,11          <181>
160 PRINTTAB(18)"{DOWN WHITE S+ C- S-}π         <232>
170 PRINTTAB(17)"{C* SHIFTSPACE CK CI CT C@     <88>
180 REM ********* ^ SHIFT SPACE *******          <158>
190 PRINT"{DOWN2 ORNG SPACE SU3 SPACE SU SPACE S
U3 SPACE SU SPACE SU SPACE2 SU3 SPACE SU3 SPACE
SU3 SPACE SU2 SPACE2 SU2 SPACE2 SU SPACE SU     <51>
200 PRINT"{SPACE SU SPACE SU SPACE SU SPACE SU S
PACE3 SU SPACE SU SPACE2 SU SPACE3 SU SPACE SU S
PACE SU SPACE3 SU SPACE SU SPACE SU SPACE SU SPA
CE SU SPACE SU                                  <21>
210 PRINT"{SPACE SU3 SPACE SU SPACE SU SPACE3 SU
3 SPACE2 SU3 SPACE SU3 SPACE SU3 SPACE SU SPACE
SU SPACE SU SPACE SU SPACE SU SPACE SU          <133>
220 PRINT"{SPACE SU2 SPACE2 SU SPACE SU SPACE3 S
U SPACE SU SPACE2 SU SPACE3 SU2 SPACE2 SU SPACE3
 SU SPACE SU SPACE SU SPACE SU SPACE2 SU        <87>
230 PRINT"{SPACE SU SPACE SU SPACE SU SPACE SU3
SPACE SU SPACE SU SPACE2 SU SPACE3 SU SPACE SU S
PACE SU3 SPACE SU2 SPACE2 SU2 SPACE3 SU         <241>
240 PRINTTAB(6)"{DOWN2 BLACK}COPYRIGHT{SPACE}198
6{SPACE}BY{SPACE WHITE}ANGEL-SOFT               <100>
250 PRINTTAB(9)"{DOWN3 BLACK SG                 <220>
260 PRINTTAB(9)"{L.BLU SE SF SPACE18 SE SF}"    <175>
270 PRINTTAB(9)"{BLACK SH                       <15>
280 PRINTTAB(8)"{DOWN3 WHITE}PRESS{SPACE BLACK}F
IRE{SPACE WHITE}TO{SPACE}START{SPACE}GAME       <185>
290 FORL=1TO8:READT:POKE818+L,T:NEXT            <74>
300 DATA65,66,67,68,77,78,79,89                 <109>
310 POKEV+6,27                                  <217>
320 T$="{SPACE17 SV SW SPACE}RICH{SPACE}FREDDY{S
PACE}IS{SPACE}WRITTEN{SPACE}BY{SPACE}O.DANGEL{SP
ACE}AND{SPACE}(C)1986{SPACE}BY                  <52>
330 T$=T$+"{SPACE}ANGEL-SOFT{SPACE SV SW SPACE}U
SE{SPACE}JOY{SPACE}1{SPACE}TO{SPACE}PLAY{SPACE}B
UT{SPACE}DON'T{SPACE}FORGET{SPACE}TO{SPACE}LOOK{
SPACE}OUT                                       <44>
340 T$=T$+"{SPACE}FOR{SPACE}OTHER{SPACE}GAMES{SP
ACE}FROM{SPACE}ANGEL-SOFT{SPACE SV SW SPACE}"   <19>
350 FORL=1TO166:SYS15360                        <254>
360 M$=MID$(T$,L,18)                            <140>
370 CHAR,11,17,"{CYAN}"+M$                      <156>
380 FORM=0TO15                                  <59>
390 IFJOY(1)=128THEN410                         <66>
400 NEXTM,L:GOTO350                             <118>
410 BI=1:TR=5:POKE252,0                         <40>
420 SCNCLR:POKEV+6,11                           <41>
430 GOSUB780:ONBIGOSUB880,1130,1400,1660:POKE208
,P                                              <74>
440 FORL=214TO222STEP4:POKEL+2,PEEK(L):POKEL+3,P
EEK(L+1):NEXT:POKEV+6,27                        <208>
450 POKE4,251:POKE209,0:POKE253,0:POKE254,0     <50>
460 VOL6:FORL=680TO780STEP2:SOUND3,L,2:NEXT:FORL
=1TO300:NEXT:VOL2                               <162>
470 REM *** MAIN LOOP ****************          <207>
480 SYS16107:ONPEEK(209)GOTO570:SOUND3,780,D    <186>
490 IFBI=3THENSYS15629:ONPEEK(209)GOTO570       <70>
500 SYS15360:SYS15570:ONPEEK(209)GOTO570        <227>
510 SYS15620                                    <144>
520 IFBI<>1THENSYS15747:ONPEEK(209)GOTO570      <248>
530 IFPEEK(254)=1THENSYS15830:ONPEEK(209)GOTO570 <249>
540 ONPEEK(198)GOTO750                          <84>
550 GOTO480                                     <129>
560 REM *** FAILED ******************           <215>
570 SOUND3,500,260:FORL=8TO1STEP-1:VOLL:SYS15423
:NEXT:PRINT"{HOME}"                             <57>
580 TR=TR-1:IFTR<>0THEN420                      <212>
590 REM *** GAME OVER ****************          <9>
600 PRINT"{HOME}"TAB(9)"{DOWN6 SPACE22}"        <75>
610 PRINTTAB(9)"{ORNG SPACE2 SU4 SPACE2 SU2 SPAC
E2 SU SPACE3 SU SPACE SU3 SPACE}"               <177>
620 PRINTTAB(9)"{SPACE SU SPACE5 SU SPACE2 SU SP
ACE SU2 SPACE SU2 SPACE SU SPACE3}"             <90>
630 PRINTTAB(9)"{SPACE SU SPACE SU3 SPACE SU4 SP
```

```
ACE SU SPACE SU SPACE SU SPACE SU3 SPACE}"                <6>
640 PRINTTAB(9)"{SPACE SU SPACE3 SU SPACE SU SPA
CE2 SU SPACE SU SPACE3 SU SPACE SU SPACE3}"               <237>
650 PRINTTAB(9)"{SPACE2 SU3 SPACE2 SU SPACE2 SU
SPACE SU SPACE3 SU SPACE SU3 SPACE}"                      <99>
660 PRINTTAB(9)"{SPACE22}"                                <134>
670 PRINTTAB(9)"{SPACE2 SU3 SPACE2 SU SPACE3 SU
SPACE SU3 SPACE SU3 SPACE2}"                              <49>
680 PRINTTAB(9)"{SPACE SU SPACE3 SU SPACE SU SPA
CE3 SU SPACE SU SPACE3 SU SPACE2 SU SPACE}"               <158>
690 PRINTTAB(9)"{SPACE SU SPACE3 SU SPACE SU SPA
CE3 SU SPACE SU3 SPACE SU3 SPACE2}"                       <232>
700 PRINTTAB(9)"{SPACE SU SPACE3 SU SPACE2 SU SP
ACE SU SPACE2 SU SPACE3 SU SPACE SU SPACE2}"              <253>
710 PRINTTAB(9)"{SPACE2 SU3 SPACE4 SU SPACE3 SU3
 SPACE SU SPACE2 SU SPACE}"                               <123>
720 PRINTTAB(9)"{SPACE22}"                                <194>
730 FORL=0TO3000:NEXT:GOTO1940                            <173>
740 REM *** NEXT LEVEL ****************                   <179>
750 BI=BI+1:IFBI<>5THEN420                                <14>
760 IFPEEK(252)=0THENBI=1:GOTO420                         <219>
770 GOTO1940                                              <91>
780 PRINTTAB(2)"{WHITE}MONEY:{CYAN}"USING"###";P
EEK(252);                                                 <31>
790 PRINTTAB(13)"{CYAN S+ C- S-}π{YELLOW C* SHIF
TSPACE CK CI CT}";                                        <22>
800 PRINTTAB(25)"{WHITE}TRIES{SPACE}LEFT:{CYAN}"
USING"##";TR                                              <195>
810 PRINT"{YL.GRN SM SN38 SO}"                            <51>
820 FORL=0TO20:PRINT"{SM}"TAB(39)"{SO}":NEXT              <66>
830 PRINT"{SM SN38 SO HOME}"                              <134>
840 CHAR,35,22,"{L.BLU SE SF}":POKE247,35:POKE24
8,22:POKE249,35:POKE250,22                                <54>
850 CHAR,37,21,"{L.BLU SA SB                              <80>
860 CHAR,37,22,"{SC SD}":RETURN                           <92>
870 REM *** SCREEN 1 ******************                   <39>
880 PRINT"{HOME RIGHT DOWN2 BLACK SK CYAN SP SPA
CE34 SP BLACK SL                                          <123>
890 PRINT"{RIGHT YL.GRN SN25 SO SPACE2 SM SN SO
SPACE2 SM SN                                              <41>
900 PRINT"{RIGHT BLACK SK SPACE9 YL.GRN SM BLACK
 SK CYAN SP SPACE3 YL.GRN SM BLACK SK CYAN SP SP
ACE3 YL.GRN SM BLACK SK CYAN SP SPACE11 SP BLACK
 SL                                                       <136>
910 PRINT"{RIGHT YL.GRN SO SPACE WHITE SV SN SPA
CE YL.GRN SM SN SO SPACE2 SM SN2 SO SPACE2 SM SN
2 SO SPACE2 SM SN2 SO SPACE2 SM SN SO SPACE2 SM
SN                                                        <219>
920 PRINT"{RIGHT BLACK SK SPACE5 L.BLU SI SPACE2
9 CYAN SP BLACK SL                                        <135>
930 PRINT"{RIGHT YL.GRN SN25 SO SPACE2 SM SN SO
SPACE2 SM SN                                              <81>
940 PRINT"{RIGHT SPACE10 YL.GRN SM SO SPACE24 CY
AN SP BLACK SL                                            <1>
950 PRINT"{RIGHT SPACE10 YL.GRN SM SO SPACE21 SM
 SN                                                       <94>
960 PRINT"{RIGHT SPACE CYAN SP SPACE L.BLU SI             <43>
970 PRINT"{RIGHT YL.GRN SN11 SO SPACE2 SM SN15 S
O                                                         <151>
980 PRINT"{RIGHT SPACE10 SM SO SPACE18 SM SO              <169>
990 PRINT"{RIGHT SPACE2 CYAN SP SPACE8 YL.GRN SM
 SO SPACE9 CYAN SP SPACE8 YL.GRN SM SO                    <213>
1000 PRINT"{RIGHT SPACE12 SM SO SPACE15 CYAN SP
SPACE2 YL.GRN SM SO                                       <114>
1010 PRINT"{RIGHT SN8 SO SPACE4 SM SO SPACE9 CYA
N SP SPACE6 YL.GRN SM SO                                  <55>
1020 PRINT"{RIGHT SPACE30 YL.GRN SM SO                    <162>
1030 PRINT"{RIGHT SN10 SO SPACE7 SM SN11 SO SPAC
E3 SM SN                                                  <40>
1040 PRINT"{RIGHT SPACE10 SM SN2 SO SPACE SM SN2
 SO BLACK SK CYAN SP                                      <173>
1050 PRINT"{RIGHT SPACE13 YL.GRN SM SN SO SPACE2
 SM SN5 SO                                                <66>
1060 PRINT"{RIGHT SPACE CYAN SP SPACE L.BLU SI S
PACE20 YL.GRN SM SN SO                                    <38>
1070 PRINT"{RIGHT SN15 SO SPACE10 SM SN SO                <1>
1080 P=50:D=7                                             <159>
1090 POKE214,8:POKE215,6                                  <223>
1100 POKE218,5:POKE219,10                                 <114>
1110 POKE222,5:POKE223,20:RETURN                          <130>
1120 REM *** SCREEN 2 ****************                    <107>
1130 PRINT"{HOME YL.GRN DOWN                              <49>
1140 PRINT"{RIGHT SN SO SPACE12 SM SO SPACE2 BLA
CK SY YL.GRN SN8 SO SPACE2 SM SN2 SO SPACE SM SN          <190>
1150 PRINT"{RIGHT SPACE SM SO SPACE6 CYAN SP SPA
CE8 BLACK SY YL.GRN SN SO CYAN SP SPACE YL.GRN S
M SN3 SO CYAN SP2 YL.GRN SM SN SO SPACE CYAN SP
SPACE YL.GRN SM SN                                        <46>
1160 PRINT"{RIGHT SPACE2 SM SO SPACE10 SM SO SPA
CE2 BLACK SY YL.GRN SN SO CYAN SP SPACE4 YL.GRN
SM SO SPACE2 SM SO SPACE5 SM                              <243>
1170 PRINT"{RIGHT SPACE CYAN SP SPACE YL.GRN SM
SO SPACE13 BLACK SY YL.GRN SM SO CYAN SP SPACE L
.BLU SI                                                   <87>
1180 PRINT"{RIGHT SPACE4 YL.GRN SM SO SPACE8 SM
SO SPACE2 BLACK SY YL.GRN SN9 SO SPACE3 SM SO             <196>
1190 PRINT"{RIGHT L.BLU SI SPACE17 BLACK SY YL.G
RN SN SO CYAN SP2 SPACE4 YL.GRN SM SO SPACE2 SM
SN2 SO                                                    <170>
1200 PRINT"{RIGHT SN12 SPACE2 SM SO SPACE2 BLACK
 SY YL.GRN SN SO BLACK SG2 SPACE4 YL.GRN SM SO S
PACE SM SO SPACE2 SM SO                                   <42>
1210 PRINT"{RIGHT BLACK SG12 SPACE6 BLACK SY YL.
GRN SN SO                                                 <115>
1220 PRINT"{RIGHT SPACE18 BLACK SY YL.GRN SN SO
BLACK SH2 SPACE4 YL.GRN SM SO SPACE SM SO SPACE2
 SM SO                                                    <212>
1230 PRINT"{RIGHT SPACE CYAN SP SPACE2 YL.GRN SM
 SO SPACE2 SM SO SPACE2 CYAN SP SPACE YL.GRN SM
SO SPACE2 BLACK SX YL.GRN SN SO CYAN SP2 SPACE4
YL.GRN SM SO SPACE2 SM SN2 SO                             <151>
1240 PRINT"{RIGHT SPACE3 SM SO SPACE4 SM SO SPAC
E7 BLACK SY YL.GRN SN9 SO SPACE3 SM SO                    <168>
1250 PRINT"{RIGHT SN3 SO SPACE2 CYAN SP2 SPACE2
YL.GRN SM SO SPACE2 SM SO SPACE2 BLACK SY YL.GRN
 SN SO WHITE SV SM SPACE4 YL.GRN SM SO                    <255>
1260 PRINT"{RIGHT SPACE3 SM SO SPACE4 SM SO SPAC
E7 BLACK SY YL.GRN SN SO BLACK SG2 SPACE3 YL.GRN
 SM SN SO SPACE2 SM SN3 SO                                <52>
1270 PRINT"{RIGHT SPACE CYAN SP SPACE2 YL.GRN SM
 SO SPACE2 SM SO SPACE2 CYAN SP SPACE YL.GRN SM
SO SPACE2 BLACK SY YL.GRN SN SO L.BLU SI                  <58>
1280 PRINT"{RIGHT SPACE18 BLACK SY YL.GRN SN16 S
O                                                         <223>
1290 PRINT"{RIGHT SPACE2 SM SN17 SO                       <203>
1300 PRINT"{RIGHT SPACE19 SM SO SPACE CYAN SP YL
.GRN SPACE3 SM SO                                         <45>
1310 PRINT"{RIGHT SN16 SO SPACE2 SM SO SPACE4 SM
 SN2 SO                                                   <53>
1320 PRINT"{RIGHT SPACE19 SM SO SPACE3 SM SO SPA
CE2 SM SO                                                 <162>
1330 PRINT"{RIGHT SPACE2 SM SN21 SO SPACE4 SM SN
3 SO                                                      <89>
1340 PRINT"{RIGHT                                         <117>
1350 P=100:D=8                                            <2>
1360 POKE214,2:POKE215,7:POKE232,19                       <168>
1370 POKE218,25:POKE219,5:POKE233,11                      <118>
1380 POKE222,23:POKE223,15:RETURN                         <223>
1390 REM *** SCREEN 3 ****************                    <133>
1400 PRINT"{HOME YL.GRN DOWN                              <64>
1410 PRINT"{RIGHT SPACE8 SM SO SPACE3 SM SO SPAC
E3 SM SO SPACE3 SM SO SPACE2 SM SN                        <104>
1420 PRINT"{RIGHT SPACE2 L.BLU SI SPACE26 YL.GRN
 SM SN2 SO SPACE3 SM SN                                   <65>
1430 PRINT"{RIGHT SPACE CYAN SP YL.GRN SM SN4 SO
 SPACE2 SM SN SO SPACE2 SM SN SO SPACE2 SM SN SO
 SPACE2 SM SO SPACE3 SM SO SPACE2 CYAN SP SPACE2
 YL.GRN SM                                                <224>
1440 PRINT"{RIGHT SPACE2 SM SN23 SO                       <45>
1450 PRINT"{RIGHT SPACE2 SM SN4 SO SPACE2 SM SN8
 SO SPACE4 SM SN SO SPACE3 SM SN SO                       <93>
1460 PRINT"{RIGHT SPACE2 SM SN4 SO SPACE2 SM SN7
 SO SPACE3 CYAN SP SPACE2 YL.GRN SM SO SPACE5 SM
SN SO                                                     <37>
1470 PRINT"{RIGHT SPACE2 SM SN3 SO SPACE4 SM SN5
 SO SPACE3 SM SN SO SPACE2 SO SPACE7 SM SN SO             <165>
1480 PRINT"{RIGHT SPACE2 SM SN3 SO SPACE CYAN SP
2 SPACE YL.GRN SM SN5 SO SPACE2 SM SN5 SO SPACE3
 SN SPACE5 SM SN                                          <219>
1490 PRINT"{RIGHT CYAN SP SPACE YL.GRN SM SN2 SO
 SPACE6 SM SN4 SO SPACE3 SM SN4 SO SPACE2 SM SN
SO SPACE5 BLACK SY                                        <128>
1500 PRINT"{RIGHT SPACE2 YL.GRN SM SN2 SO SPACE2
 SM SO SPACE2 SM SO SPACE2 SM SN SO SPACE2 CYAN
SP SPACE2 YL.GRN SM SN SO SPACE SM SO SPACE SM S
O SPACE4 BLACK SY                                         <135>
1510 PRINT"{RIGHT SPACE2 YL.GRN SM SN SO SPACE8
SM SN7 SO SPACE3 SM SN2 SO SPACE CYAN SP SPACE Y
L.GRN SM SO SPACE3 BLACK SY                               <49>
1520 PRINT"{RIGHT SPACE2 YL.GRN SM SN SO SPACE2
SM SN2 SO SPACE2 SM SN8 SO SPACE CYAN SP SPACE Y
L.GRN SO SPACE10 BLACK SY                                 <164>
```

```
1530 PRINT"{RIGHT SPACE2 YL.GRN SM SO SPACE CYAN
 SP SPACE YL.GRN SM SN2 SO SPACE CYAN SP SPACE Y
L.GRN SM SN2 SO SPACE2 SM SO SPACE4 SO SPACE10 B
LACK SX                                          <245>
1540 PRINT"{RIGHT SPACE2 YL.GRN SM SO SPACE2 SM
SN4 SO SPACE2 SM SN3 SO SPACE4 CYAN SP SPACE YL.
GRN SM SN2 SO SPACE CYAN SP SPACE YL.GRN SM SO S
PACE3 BLACK SY                                   <140>
1550 PRINT"{RIGHT SPACE CYAN SP YL.GRN SM SPACE4
 SM SN2 SO SPACE4 SM SN3 SO SPACE4 SM SN SO SPAC
E SM SN3 SO SPACE4 BLACK SY                      <82>
1560 PRINT"{RIGHT SPACE2 L.BLU SI SPACE34 BLACK
SY                                               <186>
1570 PRINT"{RIGHT SPACE2 YL.GRN SM SN23 SO SPACE
10 BLACK SY                                      <5>
1580 PRINT"{RIGHT SPACE2 L.BLU SI SPACE34 BLACK
SY                                               <206>
1590 PRINT"{RIGHT SPACE2 YL.GRN SM SPACE6 SO SPA
CE2 SM SPACE6 SO SPACE2 SM SPACE3 CYAN SP SPACE3
 YL.GRN SM SO SPACE5 BLACK SY                    <254>
1600 PRINT"{RIGHT WHITE SV SW YL.GRN SM SO SPACE
 CYAN SP SPACE2 YL.GRN SM SO SPACE2 SM SO SPACE2
 CYAN SP SPACE YL.GRN SM SO SPACE2 SM SO SPACE5
SM SN2 SO                                        <83>
1610 PRINT"{RIGHT BLACK SH2 YL.GRN SM SN SO BLAC
K SH2 YL.GRN SM SN SO BLACK SH2 YL.GRN SM SN SO
BLACK SH2 YL.GRN SM SN SO BLACK SH2 YL.GRN SM SN
 SO BLACK SH2 YL.GRN SM SN4 SO                   <49>
1620 P=100:D=9:POKE226,30:POKE227,13             <49>
1630 POKE214,4:POKE215,3:POKE232,38              <162>
1640 POKE218,4:POKE219,17:POKE233,14             <96>
1650 POKE222,4:POKE223,19:RETURN                 <11>
1660 REM *** SCREEN 4 ****************           <159>
1670 PRINT"{HOME DOWN                            <230>
1680 PRINT"{RIGHT BLACK SG4 YL.GRN SM SO BLACK S
G3 SPACE8 SG2 SPACE18 SY                         <86>
1690 PRINT"{RIGHT SPACE2 WHITE SV SW YL.GRN SM S
O BLACK SK WHITE SV SW SPACE3 BLACK SH2 SPACE8 S
H2 SPACE2 YL.GRN SM SN3 SO SPACE6 BLACK SY       <250>
1700 PRINT"{RIGHT YL.GRN SO SPACE3 SM SN18 SO SP
ACE4 WHITE ST SPACE YL.GRN SM SO SPACE5 BLACK SY <21>
1710 PRINT"{RIGHT L.BLU SI SPACE21 YL.GRN SM SO
SPACE3 WHITE SR SS SPACE2 YL.GRN SM SO SPACE4 BL
ACK SY                                           <113>
1720 PRINT"{RIGHT YL.GRN SN2 SO SPACE3 SM SN4 SO
 SPACE5 SM SN5 SO SPACE8 SM SO SPACE3 BLACK SY   <6>
1730 PRINT"{RIGHT L.BLU SI SPACE13 CYAN SP SPACE
7 YL.GRN SM SO SPACE2 SM SN SO SPACE CYAN SP SPA
CE YL.GRN SM SN SO SPACE2 BLACK SY               <155>
1740 PRINT"{RIGHT YL.GRN SN4 SO SPACE3 SM SO SPA
CE3 SM SN SO SPACE3 SM SN3 SO SPACE2 SM SN6 SO S
PACE3 BLACK SY                                   <190>
1750 PRINT"{RIGHT L.BLU SI SPACE21 YL.GRN SM SO
SPACE2 SM SO SPACE CYAN SP YL.GRN SPACE SM SO SP
ACE4 BLACK SY                                    <205>
1760 PRINT"{RIGHT YL.GRN SN6 SO SPACE4 SM SN5 SO
 SPACE3 SM SN SO SPACE2 SM SO SPACE9 BLACK SY    <79>
1770 PRINT"{RIGHT L.BLU SPACE22 YL.GRN SM SO SPA
CE2 SM SO BLACK SH7 SPACE2 SY                    <164>
1780 PRINT"{RIGHT SPACE5 YL.GRN SM SN6 SO SPACE3
 SM SN6 SO SPACE2 SM SN7 SO SPACE2 BLACK SY      <23>
1790 PRINT"{RIGHT YL.GRN SN SO SPACE20 SM SO SPA
CE2 SM SO BLACK SG7 SPACE2 SY                    <190>
1800 PRINT"{RIGHT YL.GRN SN3 SO SPACE7 SM SN5 SO
 SPACE3 SM SN SO SPACE2 SM SO SPACE9 BLACK SX    <230>
1810 PRINT"{RIGHT YL.GRN SN5 SO SPACE14 SM SN2 S
O SPACE2 SM SO SPACE CYAN SP YL.GRN SPACE SM SN2
 SO SPACE2 BLACK SY                              <249>
1820 PRINT"{RIGHT YL.GRN SN7 SO SPACE5 SM SN SO
SPACE3 SM SN3 SO SPACE2 SM SO SPACE9 BLACK SY    <89>
1830 PRINT"{RIGHT YL.GRN SN9 SO SPACE4 CYAN SP S
PACE3 YL.GRN SM SN4 SO SPACE2 SM SN7 SO SPACE2 B
LACK SY                                          <4>
1840 PRINT"{RIGHT SG10 SPACE3 SH3 SPACE2 SG6 SPA
CE2 YL.GRN SM SO SPACE9 BLACK SY                 <205>
1850 PRINT"{RIGHT SPACE13 YL.GRN SM SN SO SPACE1
4 SM SN3 SO SPACE2 BLACK SY                      <192>
1860 PRINT"{RIGHT SPACE4 SH7 SPACE7 SH6 SPACE2 Y
L.GRN SM SO SPACE9 BLACK SY                      <25>
1870 PRINT"{RIGHT SPACE CYAN SP SPACE BLACK SH Y
L.GRN SM SN5 SO BLACK SH SPACE2 CYAN SP SPACE2 B
LACK SH YL.GRN SM SN4 SO SPACE2 SM SN3 SO        <197>
1880 PRINT"{RIGHT BLACK SH3 YL.GRN SM YELLOW}(C)
1986{YL.GRN SO BLACK SH5 YL.GRN SM SN YELLOW}A-S
{YL.GRN SN SO                                    <186>
1890 P=150:D=9:POKE226,30:POKE227,10             <9>
1900 POKE214,2:POKE215,5:POKE232,38              <191>
1910 POKE218,2:POKE219,7:POKE233,14              <181>
1920 POKE222,2:POKE223,9:RETURN                  <144>
1930 REM ****************************            <86>
1940 SCNCLR:P=PEEK(252)+TR*30                    <152>
1950 PRINTTAB(6)"{DOWN6 YELLOW}FREDDY'S{SPACE}MO
NEY{SPACE}IN{SPACE}THE{SPACE}END{SPACE}IS        <126>
1960 PRINTTAB(18)"{DOWN BLACK}"USING"###";P      <90>
1970 PRINTTAB(6)"{DOWN5 WHITE}DO{SPACE}YOU{SPACE
}WANT{SPACE}TO{SPACE}PLAY{SPACE}AGAIN{SPACE}??   <30>
1980 PRINTTAB(6)"{DOWN BLACK}YES{SPACE CYAN}<    <59>
1990 PRINTTAB(6)"{BLACK SPACE}NO":VOL4:Y=16      <93>
2000 J=JOY(1):IFJ=0THEN2000                      <215>
2010 IFJ=128THEN2090                             <63>
2020 IFJ=1ANDY<>16THENY=16:GOTO2050              <195>
2030 IFJ=5ANDY<>17THENY=17:GOTO2050              <246>
2040 GOTO2000                                    <44>
2050 CHAR,10,16,"{SPACE}"                        <89>
2060 CHAR,10,17,"{SPACE}"                        <106>
2070 CHAR,10,Y,"{CYAN}<"                         <68>
2080 SOUND1,520,5:GOTO2000                       <140>
2090 IFY=16THENRUN                               <7>
2100 SYS65526                                    <247>
  ENDE DES LISTINGS
```

HARDWARE-SERVICE

Nachdem der Tronic-Verlag mit großem Erfolg eine 64K-Speichererweiterung für den C-16/116 in einer der vorangegangenen Ausgaben von »COMPUTE MIT« angeboten hat, möchten wir auch jetzt wieder auf ein neues Hardware-Angebot hinweisen. Wir planen für den Anfang des Jahres 1987 den Ausbau einer Speichererweiterung auf 256K-Byte. Dazu müssen wir jedoch von Ihnen wissen, ob bei dem User überhaupt ein Interesse dafür vorhanden ist, die Speicherkapazität des C-16/116, C-64 und Plus 4 auf die eines ausgewachsenen Personalcomputers aufzurüsten. Von der technischen Seite sind alle Probleme gelöst, so daß eigentlich sofort mit der Serienproduktion begonnen werden könnte. Wir möchten Sie deshalb bitten uns umgehend mitzuteilen, ob und an welcher Speichererweiterung Sie interessiert sind. Die genaue Aufstellung der Erweiterungen für die einzelnen Syteme und die entstehenden Kosten können Sie der nachfolgenden Gliederung entnehmen:

C-16/116 aufgerüstet auf 256K-Byte

als Bausatz
(Lötungen müssen selbst vorgenommen werden, Bauanleitung und Zubehör wird geliefert) **223,– DM**

zum Umrüsten
(Sie senden Ihren Computer zum Umrüsten an eine Firma ein) **263,– DM**

C-64/Plus 4 aufgerüstet auf 256K-Byte

als Bausatz 173,– DM

zum Umrüsten 218,– DM

Senden Sie Ihre Mitteilung umgehend an folgende Adresse:

TRONIC-VERLAG GmbH
Hardware-Service
Postfach 870

3440 Eschwege

Bitte senden Sie keine verbindlichen Bestellungen ein. Wir möchten zunächst nur die Resonanz auf unser Angebot testen und werden dann zu einem späteren Zeitpunkt in der Zeitschrift »COMPUTE MIT« ein genaues Angebot und die entsprechenden Bestell-Modalitäten bekanntgeben.

★ Kleinanzeigen ★ Kleinanzeigen ★ Kleinanzeigen ★

CSJ
COMPUTERSOFT JONIGK

C64	Cass.	Disk
Asterix	29,--	49,--
Go for Gold	19,--	29,--
Yier ar Kung Fu	32,--	39,--
Goonies	29,---	49,--
Ping Pong	34,--	46,--
Fight Night	32,--	39,--
Mercenary Compendium	49,--	59,--
Ultima IV		69,--
C128		Disk
The Last V8		24,90
Kickstart		24,90
The Rocky Horror Show		49,--
Vizawrite Classic (Deutsch)		348,--
Vizastar 128 (Deutsch)		398,--
AMIGA		Disk
Hacker II		89,--
Wisbringer		98,--
Leader Gotas		109,--
Vizawrite AMIGA		598,--
256 KB Systemspeichererweit.		298,--

C16/Plus 4	Disk	Cass.
über 150 versch. Spiele auf Lager		
17 versch. Mastertronic Games		je 9,90
Adreß, Datei, Vokabel, Text, Utility		
je Programm auf Cass.		nur DM 29,90
Danger Diamonts	39,--	29,--
Sun Street (The Newsboy)	39,--	29,--
Jump Jet (+ 16K/Plus4)		39,--
Mercenary (Plus4)		39,--
Mercenary Compendium (Plus4)		52,--
Saboteur (Plus4)		39,--
Robo Knight		9,90
Shark		9,90
Auriga		9,90
Lunar Doking		14,90
Booty		9,90
Grandmaster		29,00
Winter Events		29,00

Atari ST / IBM / Schneider auf Anfrage
Mindestbestellwert DM 30,--
Nicht vergessen CSJ NEWS anfordern

VIZA SOFTWARE
3000 Hannover 1
An der Tiefenriede 27
Tel.: Service 0511/886383
CSJ COMPUTERSOFT JONIGK

Riesenauswahl an Software
sofort CSJ NEWS anfordern
bitte Computertyp angeben
Händleranfragen erwünscht

Biete Software

"C-16/116" Verkaufe 10 TOP-GAMES (z.B. Goldrausch, Submarine, Inv. of Space, Space Ship...) für 20,-- DM auf Kassette. Info gratis. Einsenden an: Klaus-Peter Banko, Alte Poststr. 14, 4577 Nortrup, Tel: 05436/285
Suche Kontakte zwecks Erfahrungsaustausch!

C 128, C 64, C 16/116, Plus 4, VC 20
- 6 Spiele 19,90
- Textverarbeitung 12,90
- Adress, Karteikasten je 29,90
- Video-, Musikarchiv je 29,90
- Denktraining 19,90
- Vokalbeltrainer 19,90
- Katalog gegen 2x80 Pf-Marken bei: Computerservice Tino Hofstede, An der Windmühle 8, 5010 Bergheim 5

C-16/116 Software!!!
20 Spiel- und Anwenderprogramme für nur 20,-- DM. Die Programme sind auf Kassette. 20 DM als Schein oder Scheck bitte an: Daniel Beitlich, Kleine Höll-bergstr. 6, 6000 Frankfurt 50 oder Info-blatt für 80 Pf. in Briefmarken anfordern.

```
*****************
* C16+64K und Plus 4 *
* BÖRSE              *
* Ein Profi-Anwender-Programm in *
* hochauflösender Grafik, 15 Gra- *
* fiken, aktueller Kontost. u.v.m. *
* Auf Kassette für 8,50 DM + 1,50 *
* DM Porto+Versand 10 DM- *
* Schein an: Manfred Leopold, *
* Hatzfelder-Str. 116, 5600 Wup- *
* pertal 2 *
* Tel.: 0202/702876 *
*****************
```

Verkaufe über 20 Originalspiele für C16, z.B. SOCCER, SKRAMBLE, ROCKMAN, und so. Spielekassette von Robtek. Bitte melden bei René Reinthal, Tel. 02225/2890 ab 14 Uhr

Verkaufe 5 C-16/116 Spiele auf Kassette (Sherlocks, Arrow) zusammen nur 10,-DM! Vorkasse!!! Hubert Mühlbacher, Platten-berg 3, 8221 Waging

---

***** C16/116 u. + 4 GRAPHIC-ADVENTURE *****
Nach den "JÜNGERN..." jetzt das Zweite, "Die Maske" -lassen Sie sich in das Reich der Phantasie entführen. Über 45 Bilder 2x10.Min., Cassette voll be-spielt. Cassette 20,--DM (Schein) oder per NN+4,--DM VK. Bestellen bei: GBR U+K, Bickernstr. 161, 4650 Gelsen-kirchen. Info gegen 80 Pf.

C-16 SPITZENSOFTWARE
Programmpaket:
1. Pilot (brandneues Adventure)
2. Blackjack C-16
3. Flugsimulator

Cassette kommt sofort für 10,--DM Vorkasse (incl. Porto) an: Ludwig Wiesemüller, Birkenweg 37, 3408 Duderstadt 27

Brauchst Du heute noch 5 Spiele kauf bei Mini-Soft ganz viele.
Restposten: C16/116 Software
Gammelpack 1 (5 Spiele) 11,11 DM
Gammelpack 2 (10 Spiele) 18,18 DM
Die Gamespacks beinhalten: z.B. Kobra,Tennis,Moon Fighter,Jumpman u.a. Bestell es direkt bei: Mini-Soft, Stryckweg 48, 3542 Willingen

---

Achtung C-16 User!
Tausche und verkaufe Anwender- und Spielprogramme. Gr. Auswahl! Ulf Peters, Gablonzer Str. 11, 2351 Trappenkamp

---

C16, C116 Superspiele!
Info gegen Rückporto bei Hannes Kaltenbach, Prielmayerstr. 16, 7990 Friedrichshafen 1

---

15 Superspiele für C-16 auf Kassette zu verkaufen! Ab 1.1.87 bei Kai Uwe Boß, Ilmenweg 5, 6422 Herbstein. 20DM in den Umschlag stecken und abschicken !!!

C-16, C-116 + Plus 4 Superspftware, Info = 80 Pfennig. Anfordern bei Michael Husen, Ritterstr. 55, 2800 Bremen 21

```
***C 16/116/Plus 4 Zeichen- und***
***Grafikprogramme für 16K und***
***64K-Version. CAD-Basissystem***
***Info gegen Porto bei Dipl.Ing***
***M. Rätzel, Ulvenbergstr. 6,***
***D-6100 Darmstadt 13***
```

C-16 !!! Verkaufe TOP-Spiele für C-16, z.B. Bandits at Zero, Rockman usw. 20% Ermäßigung auf jedes Spiel. Liste bei Christoph Furtkamp, Tel. 02851/2511

C-16/Plus 4! Spiele und Anwenderprogramme nur 1,--DM! Große Liste anfordern gegen Briefmarke oder Rückumschlag bei Jürgen Roumen, Eichenstr. 49, 4054 Nettetal 1

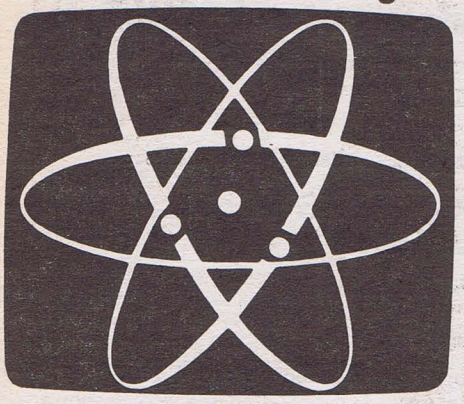
C 16/116/Plus 4:TOP-Spiele + Anwender,Graphicen + Musik zu Superpreisen. 150 Programme. Liste gegen Rückporto, DEMO-Cass. 6 DM. Ferner ab Januar 87 der Bombenhit: "USER" - Die C16 Zeitschrift auf Cassette. Mit Spielen, Anwender, Infos, Gratisanzeigen, Tips & Tricks, Preis 6 DM + 1,30 Porto. Anzeigentexte willkommen.

---

10 C-16/116 Spiele für 10,-- DM!!!
Mit: Pac-Man, Invaders of the Space, Cowboy-Duell, Tank-Wars, Turmspringen, SDI, Jäger des verlorenen Schatzes, Mimi, Cave, Submarine. Bitte frankierten Rückumschlag mit 10 DM an: Peter Preuss, Remmelsweg 5, 7107 Bad Friedrichshall 2, P.S. Tausche auch Software!

Das besondere Programm!
TURTLEGRAFIK für C-16 (64K) und Plus 4. Info gegen Rückporto bei: B. Lauer, Handgasse 10, 8700 Würzburg.

Zwei Themen - ein Ereignis:



Hobby-tronic
10. Ausstellung fur Funk- und Hobby-Elektronik

COMPUTER-SCHAU
3. Ausstellung für Computer, Software und Zubehör

Dortmund
18.–22. Februar 1987

Die umfassende Marktübersicht für Hobby-Elektroniker und Computeranwender, klar gegliedert:

In Halle 5 das Angebot für CB- und Amateurfunker, Videospieler, DX-er, Radio-, Tonband-, Video- und TV-Amateure, für Elektro-Akustik-Bastler und Elektroniker. Mit dem Actions-Center und Laborversuchen, Experimenten, Demonstrationen und vielen Tips.

In Halle 6 das Superangebot für Computeranwender in Hobby, Beruf und Ausbildung. Dazu die „Computer-Straße" als Aktionsbereich, der Wettbewerb „Jugend programmiert" und die Stände der Computerclubs.

Ausstellungsgelände Westfalenhallen Dortmund täglich 9.00-18.00 Uhr

Hardware

Verkaufe wegen Systemwechsel C116 + 64K für 150 DM. Außerdem 21 Spielkassetten, z.B. Bomb Jack, Berks, Pluspaket, Favo-rite 4, Turbo Tape u.a. für 250 DM (we-nig benutzt). Zusammen für nur 380 DM! Angebote an: Frank Roulands, Benrader-str. 135, 4154 Tönisvorst 1

Verkaufe Plus 4 nur 320,--DM, VC-20 nur 100,--DM, 16K-RAM 60,--DM, verkaufe für C-16: Gehäuse + Tastatur 25,-- DM, TED-Chip 50,-- DM, ROM-Chips je 15,--DM, Adress-Chip 15,--DM, Netzteil 15,--DM. Alle Preise + 6,--DM NN. Suche für Plus 4 EPROM-Brenner + Plotter 1520 je 130,--DM, evtl. Tausch. Christoph Drube, Kasseler Str. 28, 3530 Warburg, Tel.: 05641/1332

C16/116 Speichererweiterung 64K
- Umrüstservice DM 74,--
- Steckbar ohne löten DM 65,--
- Bausatz kompl. DM 45,--
- Datasetteninterface DM 79,--

Preise + 6,--DM Versandspesen, Info gegen Freiumschlag, Tel.: 06203/43468 (14-17 Uhr)

+ + + Verkaufe C-116 + + +
Mit BASIC-Kurs, sowie Datasette und 31 Spiele: z.B. ACE, Bombjack, Reachhead, und viele mehr. Wo? Rainer Hüttinger, Kremser Str. 14, 8500 Nürnberg 50

Verkaufe Commodore C16 + Data-sette und den Spielen Winter Olympiade, Space Pi-lot, Daley Thompsons Star Events, Plus Paket, Basic-Lernkassette und Joystick- adapter. Dazu 3 Handbücher über den C16. Preis 250,- DM. Martin Müller, Ost-str. 1, 8391 Hintereben, Tel.:08581/ 2261

Verkaufe C16 + Datasette + 12 Spiele + Joy-stick + BASIC-Cassette + BASIC-Buch! Ia-Zu-stand. VB:280,--DM, Tel:06195/4723

Verkaufe
C-16 + Datasette + Joystick + S/W-Monitor + 7TOP GAMES (z.B. Bomb Jack) Preis 500,--DM, Tel: 05602/5425, Walde-mar Glodny, Egerlandstr. 13, 3436 Hess. Lichtenau

---

Commodore C-16/116
Speichererweiterung 64 KByte Bausatz mit allen benötigten Teilen, Schalter für 16 KByte-64 KByte umzuschalten und ausführlicher Bauanleitung für nur 49,50 DM (incl. MwSt.) per Nachnahme. Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel.: 02247/8111

Commodore C16 - 116- plus 4 Userport.... Wird einfach in das Expansionsport eingeschoben. Kompatibilität zum C64 gut möglich. 2 8Bit I/O/1Seriel-Port/2 16Bit Timer/1 Echtzeit-Uhr usw. für nur 98,-- DM (incl. MwSt) + Port per NN bei: Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel.: 02247/8111

---

schnelle Lieferung · sagenhaft günstige Preise!!!

SPIELPROGRAMME

Winter Olympiade	(K) und (D)	DM 24,90
The Way of the Exploding Fist	(K)	DM 27,90
ACE (64 K Version)	(K) und (D)	DM 32,90
Commando	(K)	DM 24,90
Yie are Kung Fu	(K)	DM 24,90
International Karate	(K)	DM 19,90
Ghost'n Goblins	(K)	DM 27,90
Projekt Nova	(K)	DM 22,90
Bomb Jack	(K)	DM 19,90
Frank Bruno's Boxing	(K)	DM 19,90
Plus Paket (4 Programme)	(K) und (D)	DM 29,90
European Games (Sportspiel)	(K)	DM 19,90
Trail Blazer	(K)	DM 21,90
Computer Hits 10 Vol. 3	(K)	DM 27,90
Robo Knight	(K)	DM 9,90
Legionaire	(K)	DM 15,90
Mercenary (64 k!)	(K)	DM 26,90
Favourite 4 (4 Programme)	(K)	DM 24,90
Classics I + II (je 4 Programme)	(K) je	DM 27,90
King Size (50 Programme)	(K)	DM 24,90
Lawn Tennis	(K)	DM 24,90
Video Poker	(K)	DM 9,90
Quiwi (Trivia)	(K) und (D)	DM 23,90

ANWENDERPROGRAMME

Musik Master	(K) und (D)	DM 22,90
Micro Kalk	(K) und (D)	DM 22,90
Micro Datei	(K) und (D)	DM 22,90
Paint Box	(K) und (D)	DM 19,90
Graphic Designer	(K)	DM 15,90
Turbo Tape	(K)	DM 15,90
C-16 Utilities	(K)	DM 27,90

HARDWARE UND ZUBEHÖR

ROM Modul 64 K (Einbau ohne löten)		DM 109,00
Abdeckhaube C-16 hartplastik		DM 12,00
Abdeckhaube C-16 weichplastik		DM 9,00
Abdeckhaube Floppy 1551/1541		DM 9,00
Abdeckhaube Datasette 1530/1531		DM 6,90
Datencassetten C-15	10 Stück	DM 15,90
Disketten No Name 2D 5,25	10 Stück	DM 14,90
Thermodrucker TXP 1000 (auch f. Normalp.		DM 398,00
Farbmonitor Philips CM 8500		DM 498,00
Diskettenbox f. 100 Disk 51/4 aufkl., Schl.		DM 24,90
Joystick Quickshot II + (mit Microsch.)		DM 33,90
Joystick Quickshot II		DM 15,90
Joystickadapter		DM 9,90

Katalog mit Preisen und Programmbeschreibungen gegen DM 0,80 in Briefmarken. Versand erfolgt gegen Vorauskasse oder Nachnahme (zzgl. DM 2,50 + Zahlkartengebühr). An Versandkosten fallen an:
Bis DM 50,– 2,50; Bis DM 100,– 3,90; Bis DM 300,– 5,90; Über DM 300,– 8,50

C-16 großer Bruder des C-64
Datenfernübertragung wird zum Kinderspiel! Erweiterungsplatine >USMO I< für C-16/116 mit **RS-232 und zwei Userports** für **DM 138,--. Modem steckfertig** mit passender Software:
* Wahl von der Computer-Tastatur
* Autoanswer
* Mailboxbetrieb **DM 148,--**

Speichererweiterung DM 49,--
Guss Data Connection,
Tel.: 02723/6679

PLUS 4 + + PLUS 4 + + PLUS 4 + + Datenfernübertragung jetzt nur ein Kinderspiel. DFÜ **Komplettanlage** bestehend aus Adapter, Modem und Supersoftware. Wählen von der Computertastatur:
* Wahlwiederholung
* Auto-Answer
* Mailboxbetrieb ****229,--DM****

Ihr DFÜ-Spezialist
GUSS DATA CONNECTION, Tel.: 02723/6679

So bestellen Sie:

Der Mindestbestellwert von Software liegt bei 15.00 DM!

Sie können das Geld in bar oder per Scheck Ihrer schriftlichen Bestellung beilegen

Bei Nachnahme Versand werden 4.00 DM für Porto Kosten erhoben.

Auslandsbestellungen werden nur per Vorauskasse entgegengenommen!

Rufen Sie uns doch einfach einmal an! Unsere Telefone sind wochentags ab 9.30 UHR für Sie zu erreichen!

SOFTWARE-VERSAND
ANDREAS BACHLER
POSTFACH 429
D-4280 BOCHOLT

Wir haben die neueste Software!

Commodore 16/116/PLUS 4

	Kassette
ACE	32,00 DM
ACE (+64K)	34,95 DM
Baby Berks	9,95 DM
Bandits at Zero	14,95 DM
Bench Head	24,95 DM
Bomb Jack	29,00 DM
Classics (4 Spiele)	34,95 DM
Classics II (4 Spiele)	34,95 DM
Classics III	32,95 DM
Commando	26,95 DM
Computer-Hit (10 Spiele)	32,95 DM
Daily Thompson	25,00 DM
Finders Keepers	9,95 DM
Formula One Sim.	9,95 DM
Frank Bruno's Boxing	25,00 DM
Ghosts'n'Goblins	29,00 DM
G-Man	9,95 DM
Gull Wring Falcon	14,95 DM
International Karate	19,95 DM
Jack Attack (Modul)	14,95 DM
Jet Set Willy	25,00 DM
Kikstart	9,95 DM
Legionär	19,95 DM
Matrix/Laser Zone	29,00 DM
Monty on the Run	9,95 DM
Out on a limb	19,95 DM
Paint Box	25,00 DM
Pod	9,95 DM
Project Nova	25,00 DM
Quiwi (+64K)	29,00 DM
Raider	9,95 DM
Rockman	9,95 DM
ROM-Listing	29,00 DM
Scramble	14,95 DM
Space Pilot	19,95 DM
Speed King	9,95 DM
Street Olympics	9,95 DM
Text Manager (Disk)	38,95 DM
Text Manager (Kass.)	38,95 DM
Thai Boxing	19,95 DM
Trail Blazer	19,95 DM
Turbo Tape	18,95 DM
Water Grand Prix	9,95 DM
Wimbledon	29,00 DM
Winter Olympiade	29,00 DM
Yie Ar Kung-Fu	29,00 DM

Fordern Sie kostenloses Informationsmaterial über unser Liefer-Programm für folgende Computer an:

COMMODORE 64/128, COMMODORE 16/116/PLUS 4, SCHNEIDER 464/664
ATARI 130XE/260ST/520ST/800XL, SINCLAIR SPECTRUM und MSX

Bitte geben Sie immer Ihren Computertyp an!

TELEFON-NR. (02871) 183088

Rennersoft-Computer-Distribution
HARDWARE – SOFTWARE – PERIPHERIE
COMPUTERSPIELE für Commondore C16/116+4
z.B. Commando, Bomb-Jack, Airwolf, Legionaire, Frank Brunos Boxing, Schach, Atlantis usw.
Liste M 1 anfordern!
Telefonservice täglich von 19-22 Uhr
Postfach 920 · D-4440 Rheine · Tel.05971/81592

C-16/116/+4 Hardware & Zubehör

64K-Erweiterung & Schalter	49,50
64 K-Erweiterung o. Schalter	39,50
64 K-Erweiterung mit Einbau	65,50
Daten-Cassetten C15 5 Stück	7,80
Computape Dataset C15	11,50
Disketten Pen set 3 Stück	8,20
Endlospapier 1000 Blatt	28,50
Kunststoff-Pinzette 125 mm	0,80
Disketten 5 1/4" 10 Stück	12,80
Kopiehalter Format A4	47,50
Mikro-Lötkolben 12V/8W	9,50
Daten Cassetten C15 1 Stück	2,20
Cassetten-Archiv für 24 Stück	6,50
EDV-Endlospostk. 250 Stück	22,50

Versand per Nachnahme
Fordern Sie bitte unseren Hauptkatalog an gegen 80Pf Porto.
PC-HC-ELEKTRONIC-TECHNIK
Fritz-Reuter-Str. 3, 2914 Barßel

Suche

+ + + Achtung C-16/116 USER + + +
Tausche Original-Software
Suche C-64 zahle bis 150,-- DM
mit Zubehör auch mehr! MELDEN bei:
Guido Lange, Bergstr. 20,
8741 Schönau

Suche Tauschpartner für C-16/+4 Software! Frank Stefan, Ortelsburger Straße 23, 3400 Göttingen

C16-Einsteiger sucht Kontakte mit Profi, suche auch billig Drucker. Wer hilft mir? Antwort bitte an: A.Z. Duda, Barbarossaring 32, 6750 Kaiserslautern, Tel.: 0631/40656 nur SA-SO

Suche: C 16 oder C 116 mit Datasette und Joystick und Spielen. Angebote an: Martin Krafft, Tel.: 0231/679435 ab 14 Uhr. Preis: bis 200,-- DM

Suche C-16 ohne Speichererweiterung bis 100,-- DM sowie Drucker für C-16 bis 200,-- DM. Angebote an: Dietmar Blom, Konenhoek 13, 4407 Emsdetten

I am searching for: Genesis, biete 10,00 DM, Grandmaster, Football-Manager, Winter-Olympiade, Turbo-Tape, etc. für C-16 und Tape! R. Loepki, Heidrehmen 20, 2000 Hamburg 55

****** AN ALLE C-16-FREUNDE ******
Suche Software! Tausche auch Kiste meiner Programme gegen Rückporto an Guido Bender, Maienstr. 8, 6602 Saarbrücken-Dudweiler

Suche Kopierprogramm für C 16/64K um Progr. von Cass. auf Disk zu kopieren. H. Schrimpf, Eikeloher Str. 4, 4782 Erwitte

Verschiedenes

+ + + Achtung C-16/116 USER + + +
Tausche Original-Software
Suche C-64 zahle bis 150,-- DM
mit Zubehör auch mehr! MELDEN bei:
Guido Lange, Bergstr. 20,
8741 Schönau

Tausche Software aller Art.
Bitte meldet Euch (neue und alte Software). Daniel Schmidt; Sindelenstr. 2, CH-8340 Hinwil

C-16/116/+4 User-Club für Anfänger und Fortgeschrittene! Listings, Kurse, Sammelbestellung u.v.m. Info (50 Pf. Porto) bei Volker Huppert, Ferlingsweg 15, 4150 Okrefeld 1!

Größter C 16/116/P4 CLUB
Bietet einen tollen Service u. a.: Fachzeitschrift auf Datenträger! Informationen gegen 1,30 DM Rückporto. Hacker, Im Winger 10, 5440 Mayen 14

Münster K2-Commodore C16-CLUB
sucht Mitstreiter!
* Selbstgestrickte Software * Tausch * 64K Erweiterung * Erfahrungsaustausch * Beratung. Info: 02501-58678

C 16 wer tauscht mit mir Spiele?
Habe z.B. Tom, Schach, Baseball,usw. Schickt Eure Listen an: Kai Oertel, Suderstr. 9b, 2222 Marnerdeich, Tel.: 04851/2252 (nur Tape)

Hallo, C-16-Besitzer! Wer hat Lust mit mir Spiele zu tauschen? Schreibt mir doch einfach einmal. Holger Greve, Itzehoerstr. 15, 2351 Gnutz

Verkaufe Input 64 Ausgaben 6/85 bis 9/86 (16 Stück), nur alles zusammen, gegen Höchstgebot! Nur schriftliche Angebote an: Bernhard Helle, Schützenstr. 11, 4100 Duisburg 14

***** SCHWEIZ *****
Verkaufe DATA-BECKER-Software und Bücher mit 30%.
Tel. 01 362 88 78 (9-12/18-20 Uhr)

Ich suche die Zeitschriften 2/86, 3/86 von Compute mit.
Sven Mußwitz, An der Naßburg 17, 6369 Nidderau 1

Auftrag für Gelegenheitsanzeigen in „Compute mit ..."

An „Compute mit ..."
Tronic-Verlag
Postfach
3440 Eschwege

Unter der Rubrik „Kleinanzeigen"-veröffentlichen wir Gelegenheitsanzeigen für Verkaufsangebote, Kauf- und Tauschgesuche, Kontaktaufnahme bzw. Erfahrungsaustausch usw.
Unser Preis für „Kleinanzeigen":
Private Kleinanzeigen
bis 5 Zeilen nur DM 5,– DM
bis 10 Zeilen nur DM 10,– DM
Jede gewerbliche Gelegenheitsanzeige
bis 10 Zeilen nur 15,– DM
(dürfen nicht unter Chiffre erscheinen)
Eine Veröffentlichung erfolgt **nur** gegen Vorkasse!

Kleinanzeigen jetzt noch preiswerter

Name und Adresse _____

_____

Unterschrift _____ Datum _____

Ich wünsche folgenden Text zu veröffentlichen:

5,–

10,–

Zutreffendes ankreuzen

○ Biete Hardware ○ biete Software ○ suche ○ Kontakte ○ Versch.

KINGSOFT

präsentiert:

Neue Meisterwerke Deutscher Autoren

WINTER OLYMPIADE

von Udo Gertz

Das Spiel der Spiele für alle Besitzer eines C-116, C-16 oder Plus/4.

Ein tolles Sportfest für 1 bis 4 Mitspieler mit 6 Disziplinen, Eröffnungsfeier, Wahl der Landesfarben & Hymnen, usw.

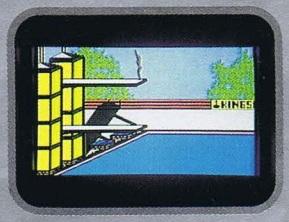
Mit diesem Programm hat Udo Gertz neue Maßstäbe gesetzt und wurde dafür soeben von der englischen Computerfachzeitschrift COMMODORE COMPUTING INTERNATIONAL mit 3 Oskars ausgezeichnet!

Und die HAPPY COMPUTER schrieb zu diesem Programm: „Was ein deutscher Programmierer da aus dem C16 herausgeholt hat, ist eine kleine Sensation. 'Winter Olympiade' ist sowohl grafisch als auch spielerisch ein Wunder. . . . 'Winter Olympiade' ist nach unserer Meinung das beste C16-Spiel, das derzeit auf dem Markt ist."

Erhältlich auf Kassette/Diskette für Commodore 116, 16, Plus/4.

WINTER OLYMPIADE

SOMMER OLYMPIADE



SOMMER OLYMPIADE

von Udo Gertz

Die Fortsetzung der sensationellen Winter Olympiade mit 6 mindestens genauso guten Disziplinen (u. a. Radfahren, Stabhochsprung, Turmspringen und Wildwasserkanu) und dem gleichen festlichen Rahmen (Eröffnungszeremonie, 1 bis 4 Mitspieler, Wahl der Landesfarben & Hymnen, usw.) – ein Muß für alle Fans von wirklich guten Sportspielen!

Erhältlich ab Januar '87 auf Kassette/Diskette für Commodore 116, 16, Plus/4.

KINGSOFT

präsentiert:

Neue Meisterwerke Deutscher Autoren

BRIDGEHEAD *von Jörg Dierks*

Der bekannte Legionär ist wieder da: diesmal soll er auf feindlichem Gebiet einen Brückenkopf errichten.
Horizontal in beide Richtungen scrollende Spitzen-Grafik mit vielen verschiedenen Bildern.

Erhältlich auf Kassette/Diskette für Commodore 116, 16, Plus/4.

KARATE KING *von Jörg Dierks*

Das endgültige Karatespiel für Ihren Plus/4 mit ausgezeichneter Animation durch neuentwickelte Multicolor-Softsprites, die flimmerfreie und schnelle Action garantieren ohne störende Farbüberschneidungen. Mehrere verschiedene Hintergrundgrafiken sorgen für fernöstliche Stimmung.
Für 1 oder 2 Spieler.

Erhältlich auf Kassette/Diskette für Commodore 16 (+64K), Plus/4.

Damit Sie sehen, daß wir außer Spielen auch was von „ernsthafter" Software verstehen, hier einige Beispiele:

BUSINESS BASIC *von Andreas Arens & Michael Meiszl*
Das „unmögliche" Modul für den C-64: 61183 Bytes frei für BASIC-Programme und -Variablen sowie über 50 neue BASIC-Befehle! ◆ *Erhältlich als Modul für C-64*

MICRO DATEI *von Jürgen Kuck*
Universelles Dateiprogramm für beliebige Daten (z. B. Adressen, Schallplatten, Videos, usw.) Leistungsmerkmale: Freier Aufbau der Eingabemaske auf dem gesamten Bildschirm, Eingeben und Ändern von Datensätzen, Sortieren nach beliebigen Kriterien, sehr komfortable Ausdruckmöglichkeiten (z. B. selektiert, mit oder ohne Maske, auf Adreßetiketten), Suchen von beliebigen Ausdrücken, 4 Funktionstasten können frei mit beliebigen Texten belegt werden (dadurch schnelle Eingabe von häufig vorkommenden Wörtern), usw.
◆ *Erhältlich auf Kassette/Diskette für Commodore 116, 16, Plus/4.*

MICRO KALK *von Fritz Schäfer*
Mit diesem Tabellenkalkulationsprogramm können Sie alle Berechnungen durchführen, die tagtäglich anfallen, z. B. Führung einer Haushaltskasse, Einkauf-/Verkauf-Erlös, usw. Leistungsmerkmale: Vielfältige mathematische Funktionen (Addition, Subtraktion, Multiplikation, Division, Summe, Durchschnitt, Minimum, Maximum), Einfügen/Löschen von Spalten/Zeilen, Ausdruck des Arbeitsblattes (ganz oder teilweise), Kopieren von Formeln und Werten in andere Zellen, 4 Funktionstasten können frei mit beliebigen Texten belegt werden (dadurch schnelle Eingabe von häufig vorkommenden Wörtern), usw. ◆ *Erhältlich auf Kassette/Diskette für Commodore 116, 16, Plus/4*

MICRO TEXT *von Fritz Schäfer*
Leistungsfähiges und einfach zu bedienendes Textverarbeitungsprogramm mit vielen Profi-Funktionen. Leistungsmerkmale: Texte werden mit integriertem Turbo-Tape auf Kassette abgespeichert, Flattersatz (linksbündig), Blocksatz (rechtsbündig), Zentrieren, Suchen und Ersetzen, halbautomatische Wort-Trennung, 4 Funktionstasten können frei mit beliebigen Texten belegt werden (dadurch schnelle Eingabe von häufig vorkommenden Wörtern) usw. Der Textspeicher umfaßt auch mit 16K bereits 6000 Zeichen (mehr als eine DIN A4 Seite).
◆ *Erhältlich auf Kassette/Diskette für Commodore 116, 16, Plus/4.*

TURBO PLUS *von Andreas Arens*
Mit diesem Modul wird Ihr C-16 bzw. Plus/4 um zahlreiche neue Funktionen erweitert, u. a.: fest eingebautes Turbo Tape (8mal schneller laden & speichern von Kassette), daß Programme bis zu 60/K abspeichern kann – zahlreiche neue BASIC-Befehle wie OLD, DUMP, WINDOW, MERGE, usw. – ein BASIC-Listing kann mit den Cursor-Tasten vor- und rückwärts gescrollt werden. Als besonderer Clou erlaubt es TURBO PLUS Besitzern eines Plus/4, die eingebaute Software auch mit Kassette zu benutzen! ◆ *Erhältlich als Steckmodul für C-116, C-16 und Plus/4.*

Sie erhalten die Spiele in den Fachabteilungen von Hertie, Kaufhof, Quelle International, Karstadt sowie in allen gutsortierten Computershops und im guten Versandhandel.
Vertrieb: **RUSHWARE** und **MICRO HÄNDLER**, in Österreich: **KARASOFT**

ACHTUNG! Wir suchen ständig Programmierer für fast alle Computer-Typen, die gegen erstklassige Bezahlung Spiele von internationalem Niveau schreiben.
Wenn Sie Interesse haben, wenden Sie sich noch heute an uns.

KINGSOFT
Spitzensoftware
Made in Germany
Seit 1983
F. Schäfer, Schnackebusch 4,
D-5106 Roetgen, Telefon (0 24 08) 51 19